Special Section: Research in Integrating Learning Capabilities into Information Systems

TING-PENG LIANG
Guest Editor

TING-PENG LIANG is a Professor and Director of the Institute of Information Management at National Sun Yat-Sen University in Taiwan, and an Associate Professor of Information Systems at the Krannert Graduate School of Management at Purdue University. Prior to joining Purdue University, he was on the faculty of the University of Illinois at Urbana–Champaign. Dr. Liang received his Ph.D. in information systems from The Wharton School of the University of Pennsylvania. His research interests include machine learning, model management, and decision support systems. He has published more than twenty papers in research journals including *Management Science, Operations Research, Journal of Management Information Systems, MIS Quarterly, Decision Support Systems, IIE Transactions*, and *Decision Sciences*, among others.

ABSTRACT: Developing intelligent information systems has been a focus of recent research. A major component that makes a system intelligent is its learning capabilities. A learning system can adapt itself to new environments and improve its performance with minimum intervention from the developer. In this paper, we review major learning paradigms, examine the role of learning in intelligent information systems, and discuss potential research issues in integrating learning capabilities in information systems design.

KEY WORDS AND PHRASES: intelligent systems, machine learning, system integration.

1. Introduction

DEVELOPING INTELLIGENT AND USER-FRIENDLY SYSTEMS HAS BEEN A FOCUS in information systems for a long time. Recently, the rapid advancement in artificial intelligence (AI) has generated many new technologies that can be integrated to enhance the performance of information systems. One field of particular potential is machine learning, a discipline that focuses on the development and analysis of learning methods suitable for computer implementation. An increasing number of articles on the development and application of machine learning techniques have appeared in information systems journals in the past several years. Like the papers included in this special section, however, most of this research is oriented toward individual application experience [4, 18, 25, 26] or method comparison [14, 20]. As the field grows, we

Journal of Management Information Systems / Spring 1993, Vol. 9, No. 4, pp. 5–15

need a general framework to consolidate existing findings and provide guidelines for future research. Toward this end, this article reviews major machine learning paradigms, examines the roles of learning in various kinds of systems, and discusses potential research issues.

Learning denotes a mechanism that enables an entity to adapt its behavior over time so that the same task or more complex tasks can be performed more efficiently or effectively [28]. The objective of machine learning is "the exploration of alternative learning mechanisms, including discovery of different induction algorithms, the scope and limitations of certain methods, the information that must be available to the learner, the issue of coping with imperfect training data, and the creation of general techniques applicable in many task domains."[3]

A question one may ask is, "Given the powerful human learning capability, why should we worry about machine learning?" There are several reasons to strive for machine learning. First, human learning is a complicated and slow process that is difficult to understand fully. It takes more than twenty years of education to train a professional physician and more than ten years of learning to make a chess master. We certainly hope that more effective learning methods can be discovered to expedite the process. Second, human learning often does not guarantee consistent performance. We can learn very complex relationships, but we often ignore the obvious ones. Third, the results from human learning are difficult to articulate, transfer, or duplicate. The same knowledge has to be learned again and again by billions of people, which is obviously inefficient. Machine learning allows knowledge to be learned once and then copied quickly for different uses. Finally, it is very difficult to integrate human learning into computer-based information systems. It is hard to interface human knowledge with computers without a substantial amount of conversion and coding. Machine learning methods can be easily integrated to make information systems more adaptive.

Another question that may be asked is, "Why do we need learning systems that integrate machine learning capabilities?" Or, in other words, what roles do learning systems play in decision making? A learning system can adapt itself to the changing environment and user needs. It plays two essential roles in decision making. First, it allows computers to be smart enough to replace human beings in certain decision situations. In fact, this has been the goal of expert systems in the past decade. Due to the difficulty of knowledge acquisition and lack of flexibility, however, the application of expert systems has faced a major crisis. Integrating learning capabilities that enable expert systems to modify their knowledge base is a key component for the second-generation expert systems.

Second, learning systems can be diligent and intelligent assistants or partners for the decision maker. The learning capability allows systems to support decision makers in a more flexible way. For example, a loan evaluation system with learning capabilities can adjust its decision criteria according to the changing risks on the market and inform the loan officer of the situation. This would have to be done manually by knowledge engineers after the decision maker discovers the need for changes. Therefore, the goals of integrating learning capabilities are twofold. First, we would like to increase the *flexibility* and *friendliness* of the system. Second, we want to improve the *performance* of the system.

Given these advantages for machine learning, it is interesting to examine how systems can learn, what should be considered when we design learning systems, and research issues involved in developing and applying learning systems.

2. How Can Information Systems Learn?

MANY METHODS HAVE BEEN DEVELOPED TO HELP COMPUTERS LEARN. Different methods often use different mechanisms to construct knowledge in different formats. In figure 1, knowledge representation, domain knowledge requirement, and learning strategies are used to differentiate learning paradigms. Regarding representation format, the knowledge learned can be *symbolic* or *nonsymbolic*. Symbolic learning systems construct if–then rules or decision trees to represent the knowledge for decision making. For example, Quinlan's ID3 [22] induces decision trees and Liang's CRIS [11] generates decision rules. Nonsymbolic learning systems construct other forms of knowledge. For example, neural networks (NN) [17] allow knowledge to be stored in a set of interconnected electronic neurons. Genetic algorithms [6] can be applied to both symbolic and nonsymbolic learning.

Learning methods can also be differentiated by the domain knowledge required for learning and their learning strategies. Some learning methods use little domain knowledge in their learning process, whereas other learning methods require extensive domain knowledge before new knowledge can be derived. For example, the back-propagation-based neural network can be applied to any problem domain with little modification. It is called a *knowledge-poor approach*. Learning by analogy [1, 2, 12] and explanation-based learning [21] require extensive domain knowledge. They are *knowledge-intensive approaches*.

Four learning strategies are popular in existing learning methods: *memorization*, *induction*, *deduction*, and *analogy*.

2.1. Learning by Memorization

This is the most primitive form of learning. The system accepts and memorizes new facts without further transformation or generalization. As the facts stored in the database increase, the performance of the system also increases. This strategy is used in several ways, including rote learning and learning by being told. The major concern in this strategy is how to index the stored knowledge properly for future retrieval. Learning takes place in the indexing process.

2.2. Learning by Induction

This is by far the most popular learning strategy. Induction is a process by which more general descriptions are created from specific instances. Learning by induction is also called inductive learning, rule induction (if rules are generated), tree induction (if decision trees are generated), or learning from examples. For example, given three products of different prices and qualities, their profits are shown as follows:

Product	Price	Quality	Profit
A	high	high	high
B	high	low	low
C	low	high	high

Two rules can be induced from the data:

Rule 1: If Quality (Product) = high, then Profit (Product) = high.
Rule 2: If Quality (Product) = low, then Profit (Product) = low.

Depending upon whether the outcomes of the input cases are known, inductive learning methods can be further divided into two categories: *supervised learning* and *unsupervised learning*. In our previous example, the outcome (profit level) of each input case is known. It is a supervised learning. Unsupervised learning classifies the input cases into clusters based on the similarity among the input attribute values and is often called *cluster analysis* or *knowledge discovery*. Supervised learning is far more popular than unsupervised learning. Most learning methods used in business domains including Quinlan's ID3, Liang's CRIS, and neural networks are supervised learning methods. This is because the mechanism of supervised learning is very similar to that of traditional discriminant analysis or statistical regression analysis used extensively in business problem solving.

2.3. Learning by Deduction

Deduction is a process by which new causal relationships are derived from old ones. For example, if we know that (1) the increased interest rate will cause higher interest payments and (2) higher interest payments will worsen cash flow, then we can deduce that the increase in interest rate will deteriorate the cash flow. A popular deduction-based approach is explanation-based learning (EBL), which uses extensive domain knowledge to produce a valid generalization of an example along with a deductive justification of the generalization [21].

The EBL process includes two stages: *explanation* and *generalization*. First, a structure of domain theories (called an *explanation*) that proves how the input example satisfies the goal concept to be learned is constructed. The explanation is then generalized by determining a set of conditions under which the explanation structure holds. For example, *X* company with a capital of 5 million, sales of 2 million, and costs totaling 8 million went bankrupt. We want to learn the rules for assessing firm bankruptcy from the example. Suppose the following domain theories are available:

T1: Profit = sales – costs;
T2: If profit < 0, then loss = absolute_value(profit);
T3: If loss > capital, then bankrupt;
T4: sales = price * volume.

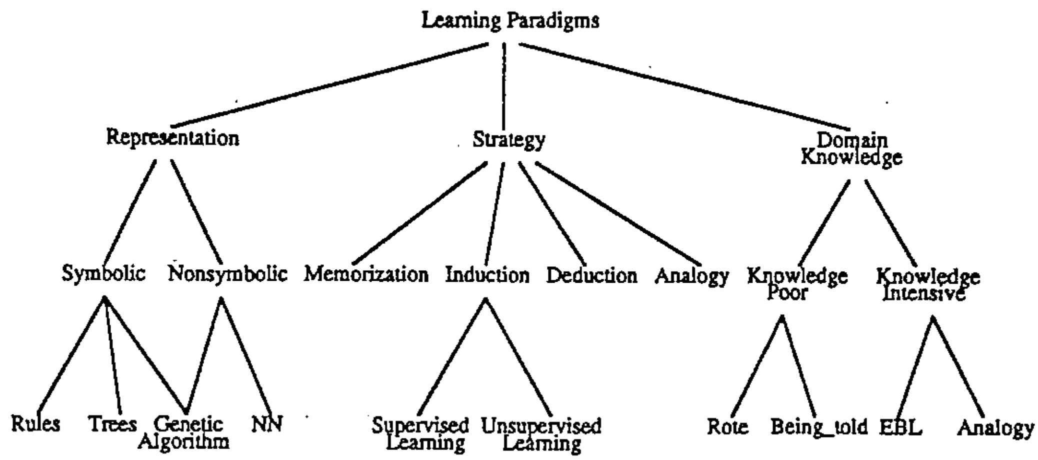

Figure 1. Classification of Learning Paradigms

Using the above information, EBL can construct an explanation structure, as shown in figure 2, from the sample firm. In the structure, three theories are used (T4 is found irrelevant in this example). To make the learned knowledge more useful, the structure can be further generalized as follows:

> If (sales – costs) < 0 and absolute_value(sales – costs) > capital,
> then bankrupt = yes.

2.4. Learning by Analogy

Learning by analogy is a process that combines both inductive and deductive learning. A typical analogical reasoning process involves (1) finding proper analogies by identifying a common substructure among descriptions from different domains, and (2) applying analogical mapping to construct new knowledge [1, 2]. For example, if curriculum design is considered analogous to a production line design, then the knowledge of production line scheduling, staff assignment, and performance evaluation can be transformed to become the new knowledge learned about curriculum design. In the analogical learning process, identifying a common substructure is an inductive process, whereas analogical mapping is a deductive process.

Analogical learning becomes more powerful when it is combined with case-based reasoning [7, 10, 24]. A case base is a repository of existing examples that have already been solved. Two kinds of learning occur in case-based systems. First, new cases can be stored in the case base and then retrieved later. This is learning by memorization. Second, knowledge can be constructed analogically using the cases stored in the case base.

3. Framework for Integrating Learning into Information Systems

WHEN WE DEVELOP LEARNING SYSTEMS, THERE ARE AT LEAST THREE ISSUES that must be considered: (1) the type of systems to be integrated, (2) the proper learning method, and (3) the integration strategy.

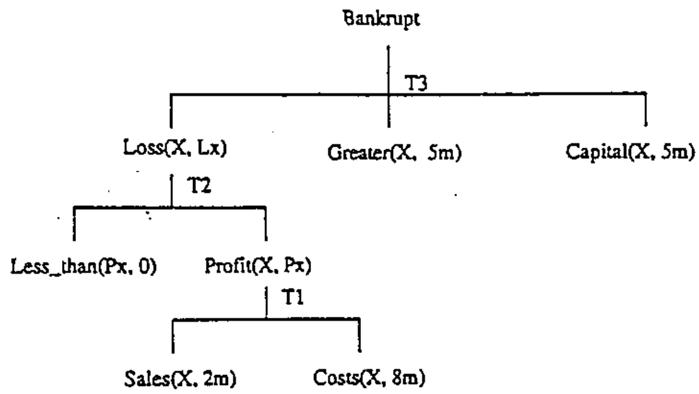

Figure 2. Explanation Structure

3.1. Type of Systems

In general, information systems can be divided into three categories: data processing systems (DPS), decision support systems (DSS), and expert systems (ES). Different types of systems have different characteristics and require different kinds of learning capabilities.

DPS are database-oriented systems. A well-known example are the airline reservation systems that maintain a large volume of travel data. Database management, data analysis, and information reporting are three major functions of DPS. Learning capabilities can be integrated into any of these three functions to make such a system more powerful. For example, data management may be improved by integrating methods of learning by memorization to allow more flexibility in indexing and retrieving key information. Data analysis capabilities can be improved by incorporating unsupervised inductive learning methods that discover customer travel patterns from database [5]. Furthermore, supervised inductive learning may help retrieve travel patterns and determine the relationships between customer profiles and travel patterns.

DSS are systems with extensive quantitative model analysis to support decision makers. Each DSS is composed of a database, a modelbase, a user interface, and a control module. Learning can be integrated into any of them. For instance, the model management function can be enhanced by including analogical and other learning methods [12, 27]. The database can also hold user profiles so that rules can be induced to make DSS self-adaptive when interacting with different users [15].

ES are systems designed to preserve human expertise. A traditional ES is composed of a knowledge base, a database, and an inference engine. Because of difficulties in knowledge engineering, learning has significant impacts on almost every phase of ES development. Inductive learning can help knowledge engineers overcome some problems in knowledge acquisition. It also allows the knowledge base to evolve as the data in the database change. Deductive learning may be integrated into the inference engine to adjust the inference strategy of the system. A casebase can also add to the ES architecture to make effective use of analogical and case-based learning. The cases in the casebase can be used to induce new knowledge, evaluate and refine existing knowledge, and provide convincing evidence to support the conclusion of the system.

3.2. Learning Technology

Given different functional modules in different types of systems, it is difficult and may be unnecessary to determine exactly which method is the most appropriate for what kind of systems. However, we can observe possible modules in which a particular method may be useful. In Table 1 we list potential opportunities for integrating machine learning into information systems.

For data processing systems, learning by memorization can make data management more flexible. Inductive learning methods such as Quinlan's ID3, neural networks, knowledge discovery, and deductive learning methods such as EBL are all useful for data analysis (such as finding patterns from the database). Knowledge discovery and analogical methods are useful for data reporting (such as finding significant information to be included in reports).

For decision support systems, learning by memorization and knowledge discovery methods can be integrated into the user interface and model utilization modules to maintain usage profiles so that the system can be more adaptive [15]. Inductive learning methods can be used to analyze model utilization and other phases of decision making to improve the management of models (including creation, selection, utilization, and other phases of model management). Neural networks and analogical learning may be useful for model construction. Deductive methods can be integrated to improve DSS control mechanisms. Analogical and case-based learning can build more intelligent what-if capabilities that allow previous experience to be included in the analysis.

For expert systems, learning by memorization can enhance the management of database and casebase. Inductive learning and knowledge discovery methods are critical to knowledge acquisition. Neural network, deductive learning, and analogical methods can be integrated to improve the inference engine. Analogical and case-based methods are also valuable for explaining system conclusions.

3.3. Integration Strategy

At least two strategies are available for the integration of learning capabilities in information systems. One is considered more *static*. One or more learning methods are built into the system and tightly coupled with the other functions of the system. A major advantage of this method is that continuous performance improvement is possible. However, a tightly integrated learning module may also result in unstable systems and increased computing costs. For instance, the knowledge base of an expert system may be changing all the time if its inductive learning module is activated every time a new case is encountered.

Another integration strategy is *dynamic*. Instead of directly linking the learning module and the system, we implement a performance monitor to examine system performance continuously. The learning module is separated from the system and is activated only if chances for a significant performance improvement are identified by the performance monitor. This strategy allows many learning methods to be built and stored in a repository similar to a database and to be shared by different systems.

Table 1 Opportunities for Integrating Learning Capabilities

Technology	DPS	DSS	ES
Memorization	Data management	User interface, model utilization	Database, casebase management
Inductive learning	Data analysis	Model management	Knowledge acquisition
Neural networks	Data analysis	Modeling	Inference acquisition
Knowledge discovery	Reporting, data analysis	User interface, database	Knowledge
Deduction	Data analysis	Control mechanism	Inference
Analogy	Reporting	Modeling, what-if analysis	Inference, explanation

In summary, there are a few issues we must consider when we develop learning systems. First, we must determine where we want the system to learn. Do we want to make the user interface more adaptive or the knowledge base more accurate? Then, we choose one or more learning methods appropriate for our purposes. For example, if we want to improve the accuracy of the rules in the knowledge base, rule induction may be a good choice. Finally, we decide on the strategy for implementation. Do we want the learning method to be tightly integrated into the system for continuous learning or loosely integrated through a performance monitor?

4. Research in Learning Systems

GIVEN THE ABOVE FRAMEWORK, A FEW RESEARCH ISSUES about developing learning systems can be identified. These include development of new learning methods, integration of different learning methods, empirical evaluation and comparison of methods, social and managerial implications of learning methods, and exploration of innovative applications. They are discussed briefly in this section.

4.1. Development of New Learning Methods

Because there are many different types of information systems designed for various tasks, no single learning method is suitable for all systems. So far, much research in machine learning has been focused on developing new and better learning algorithms that can capture the knowledge more precisely.

In general, the development of new methods is triggered by problems encountered in applying existing methods. Most existing learning paradigms have certain deficiencies. New methods often aim at alleviating these deficiencies. For instance, Quinlan's ID3 algorithm [22] accepts input data and then uses the entropy function iteratively to build a decision tree. It has problems such as: (1) the tree may be too complex if all details are covered and (2) nominal and non-nominal attributes are handled by the same criteria. As a result, modifications have been introduced to prune the tree after its construction. A method that treats nominal and nonnominal attributes differently has also been proposed [11].

In some other cases, research is triggered by the constraints of the application environment. For example, most existing learning methods assume that the system is centralized and do not consider the distributed nature of modern information systems. Therefore, developing a distributed knowledge acquisition system becomes a viable research topic, as presented in Kiang, Chi, and Tam's paper [9].

4.2. Integration of Different Methods

Since different methods have different strengths and weaknesses, another line of thought is oriented toward integrating existing methods to solve a particular problem. Method integration can be *surface* or *deep* [13]. Surface integration means different methods are coupled and the results are integrated. Deep integration means algorithms are deeply interwoven in the whole learning process. The paper by Tessmer, Shaw, and Gentry [29] is an example of method integration that integrates results from inductive learning to form more extensive knowledge. In addition to the integration of different machine learning methods, they can also be integrated with statistical and operations research methods to solve particular problems more effectively [13, 16].

4.3. Evaluation and Comparison of Methods

Choosing a proper learning method for a particular application is an important problem. It requires extensive knowledge of the performance of different methods. Therefore, research on the evaluation and comparison of learning methods is also very important. A learning method can be compared with human decision makers or other learning methods. Kattan, Adams, and Parks [8] compare machine learning tools with human judgment to find that human learning is slow and may not be better in classification tasks. In [14, 20], learning methods are compared with one another or with traditional statistical methods.

Evaluation and comparison can be performed empirically or theoretically. In empirical comparison, we need to develop test data, design experiments, run the test data with selected methods, and then compare their performance. A problem of empirical evaluation is that the results may not be generalizable in some situations. This can be alleviated by using theoretical analysis. An example of theoretical analysis can be found in [23], where O'Leary examines the impact of data accuracy on system learning to prove the importance of data accuracy in learning systems.

4.4. Social and Managerial Implications of Learning Systems

Since learning systems will, to some extent, change the way we work and make decisions, it is reasonable to expect that some impacts will be generated. Therefore, it is also interesting to examine the social and managerial implications of developing and using intelligent systems. Issues such as the following are important but significantly underexplored:

- What social impacts would learning systems generate?
- What impacts would learning systems have on human learning?
- How should we manage the development and application of learning systems?
- How can we justify the costs of development and application of learning systems?
- How can we make learning systems and human beings cooperate better?

4.5. Innovative Applications of Learning Systems

Because of the nature of machine learning, most learning applications have been related to expert systems or decision support. In fact, the learning technology may have just as much, if not more, potential in other types of systems such as transaction processing, office automation, and data processing. Therefore, finding innovative applications would be another important area for future research.

5. Concluding Remarks

DEVELOPING INTELLIGENT SYSTEMS HAS BEEN A GOAL for information systems professionals for a long time. The recent advancement in machine learning makes it possible to begin integrating learning capabilities into systems. In this article, we have reviewed major learning paradigms, presented a framework for research in this area, and pointed out five major areas for future research. Although much work has to be done before we can have a really intelligent system, now is a good time to start research.

REFERENCES

1. Burstein, M.H. Concept formation by incremental analogical reasoning and debugging. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach,* vol. 2. Los Altos, CA: Morgan Kaufmann, 1986, pp. 351–370.

2. Carbonell, J. Deviational analogy: a theory of reconstructive problem solving and expertise acquisition. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach,* vol. 2. Los Altos, CA: Morgan Kaufmann, 1986, pp. 371–392.

3. Carbonell, J.G.; Michalski, R.S.; and Mitchell, T.M. An overview of machine learning. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach.* Los Altos, CA: Morgan Kaufmann, 1983, pp. 1–23.

4. Carter, C., and Catlett, J. Assessing credit card applications using machine learning. *IEEE Expert* (Fall 1987), 71–79.

5. Frawley, W.J.; Piatetsky-Shapiro, G.; and Matheus, C.J. Knowledge discovery in databases: an overview. *AI Magazine, 14,* 3 (Fall 1992), 57–70.

6. Goldberg, D.E. *Genetic Algorithms: In Search, Optimization, and Machine Learning.* Reading, MA: Addison-Wesley, 1989.

7. Hammond, K.J. *Case-based Planning.* San Diego: Academic Press, 1989.

8. Kattan, M.W.; Adams, D.A.; and Parks, M.S. A comparison of machine learning with human judgment. *Journal of Management Information Systems, 9,* 4 (Spring 1993), this issue.

9. Kiang, M.Y.; Chi, R.T.; and Tam, K.Y. DKAS: a distributed knowledge acquisition system

in a decision support system. *Journal of Management Information Systems, 9*, 4 (Spring 1993), this issue.

10. Kolodner, J.L. Improving human decision making through case-based decision aiding. *AI Magazine, 12*, 2 (Summer 1991), 52–68.

11. Liang, T.P. A composite approach to inducing knowledge for expert systems design. *Management Science, 38*, 1 (January 1992), 1–17.

12. Liang, T.P. Analogical reasoning and case-based learning in model management systems. *Decision Support Systems, 9* (April 1993).

13. Liang, T.P.; Chandler, J.S.; and Han, I. Integrating statistical and inductive learning methods for knowledge acquisition. *Expert Systems with Applications, 1*, 4 (1990), 391–401.

14. Liang, T.P.; Chandler, J.S.; Han, I.; and Roan, J. An empirical investigation of some data effects on the classification accuracy of probit, ID3, and neural networks. *Contemporaty Accounting Research* (Fall 1992).

15. Liang, T.P., and Jones, C.V. Design of a self-adaptive decision support system. *Journal of Management Information Systems, 4*, 1 (Summer 1987), 59–82.

16. Liang, T.P.; Moskowitz, H.; and Yih, Y. Integrating neural networks and semi-Markov processes for automated knowledge acquisition: an application to real-time scheduling. *Decision Sciences* (November/December 1992).

17. McClelland, J.L., and Rumelhart, D.E. *Explorations in Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1988.

18. Messier, W.F., Jr., and Hansen, J.V. Inducing rules for expert systems development: an example using default and bankruptcy data. *Management Science, 34*, 12 (December 1988), 1403–1415.

19. Michalski, R.S. Understanding the nature of learning issues and research directions. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, vol. 2. Los Altos, CA: Morgan Kaufmann, 1986, pp. 3–25.

20. Mingers, J. An empirical comparison of selection measures for decision-tree induction. *Machine Learning, 3* (1989), 319–342.

21. Mitchell, T.M.; Keller, R.M.; and Kedar-Cabelli, S.T. Explanation-based generalization: a unifying view. *Machine Learning, 1*, 1 (1986), 47–80.

22. Quinlan, J.R. Induction of decision trees. *Machine Learning, 1*, 1 (1986), 81–106.

23. O'Leary, D.E. The impact of data accuracy on system learning. *Journal of Management Information Systems, 9*, 4 (Spring 1993), this issue.

24. Riesbeck, C.K., and Schank, R.C. *Inside Case-based Reasoning*. Hillsdale, NJ: Lawrence Erlbaum, 1989.

25. Shaw, M.J., and Gentry, J.A. Using an expert system with inductive learning to evaluate business loans. *Financial Management* (Autumn 1988), 45–56.

26. Shaw, M.J., and Gentry, J.A. Inductive learning for risk classification. *IEEE Expert* (February 1990), 47–53.

27. Shaw, M.J.; Tu, P.; and De, P. Applying machine learning to model management in decision support systems. *Decision Support Systems, 4*, 3 (1988), 285–305.

28. Simon, H.A. Why should machines learn? In R.S. Michalski, J.G. Carbonell, and T,M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*. Los Altos, CA: Morgan Kaufmann, 1983, pp. 25–37.

29. Tessmer, A.C.; Shaw, M.J.; and Gentry, J.A. Inductive learning for international financial analysis: a layered approach. *Journal of Management Information Systems, 9*, 4 (Spring 1993), this issue.