Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1994), *8*, 81-82. Printed in the USA.

GUEST EDITORIAL

Special Issue: Machine Learning in Design

MARY LOU MAHER,[1] DAVID C. BROWN,[2] AND ALEX DUFFY[3]

[1] Department of Architectural and Design Science, University of Sydney, Sydney, NSW 2006, Australia.
[2] AI Research Group, Computer Science Department, Worcester Polytechnic Institute, Worcester, MA 01609, USA.
[3] CAD Centre, University of Strathclyde, 75 Montrose Street, Glasgow, G1 1XJ, Scotland, UK.

The linking of research in machine learning with research in knowledge-based design is such that each of the two areas benefit from the consideration of the other. The use of machine learning in design addresses the perceived need to support the capture and representation of design knowledge, because handcrafting a representation is a difficult and time-consuming task. In addition, design provides a task with which to investigate the usefulness of existing machine learning techniques, and, perhaps, to discover new ones.

Knowledge-based design systems are typically encoded as compiled experience, in the form of generalized rules and/or objects. Machine learning research provides a set of theories and methods for learning from a variety of sources, such as from uncompiled experience in the form of examples of previous problems. Thus, the design knowledge need not be completely handcrafted. The knowledge learned can be used to solve problems faster, or better, or to solve problems that could not be solved before.

From the perspective of design, knowledge-based design systems provide support for a wide range of activities broadly decomposed into two tasks, synthesis and evaluation/criticism. Synthesis systems are concerned with producing a design solution or solutions from a set of specifications. Evaluation systems are concerned with determining performance or critiquing a given design. The form of knowledge needed for a synthesis system differs from that needed for an evaluation system, although both types of knowledge can be induced from design experience. The role of machine learning, and the methods which are appropriate, varies according to the type of knowledge being learned.

Reprint requests to: Dr. David C. Brown, AI Research Group, Computer Science Department, Fuller Labs 131, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609.

Dr. Mary Lou Maher: Email: mary@archsci.su.oz.au, Phone: 0011-61-2-692-4108, FAX: 0011-61-2-692-3031. Dr. David C. Brown: Email:dcb@cs.wpi.edu, Phone: (508) 831-5618, FAX: (508) 831-5776. Dr. Alex Duffy: Email: alex@cad.strath.ac.uk, Phone: +44-41-552-4400, Ext. 3005, FAX: +44-41-552-3148.

This issue of *AI EDAM* is based on a workshop on Machine Learning in Design held at the 1992 Conference on Artificial Intelligence in Design (Gero, 1992). The purpose of the workshop was to explore the issues involved in applying, extending, or developing new theories and methods for machine learning in design. In particular, it focussed on the following discussion topics:

1. Machine learning paradigms: should they guide machine learning in design?
2. Can induction be used to develop a dynamic memory of design experience/knowledge?
3. Comparing and critiquing symbolic vs. sub-symbolic learning in design.
4. Where is the greatest potential for machine learning in design?
5. Can its use in design suggest new machine learning techniques?

As a result of the workshop, a subset of the workshop papers were selected for expansion into this special issue of *AI EDAM*. The expansion was based on further development by the authors and by consideration of the workshop discussion.

The papers presented here can be considered according to the type of machine leaning employed. Machine learning methods can be classified in overlapping categories. These include: evolutionary methods, connectionist methods, inductive methods, analogical reasoning methods, and knowledge compilation methods. The evolutionary methods use genetics as an analogy for evolving knowledge. The connectionist methods are subsymbolic neural networks. The inductive methods produce a structured representation of generalized knowledge from detailed examples using various clustering/classification techniques. The analogical reasoning methods provide a set of techniques for solving a new problem using the solution to a previous, different problem. Knowing compilation methods generate new representations of knowledge from existing representations.

The *connectionist paradigm,* or neural networks, is based on an analogy between neurons in the human brain and nodes in a computer-based network. The structure and use of the network depends on the model being used. Variations occur in the number of nodes, classification of nodes as input, output, or hidden, and the procedure for allocating weights to the links in the network. Training a neural network occurs by modifying the weights of the links of the network so that a particular input consistently produces the same output. A trained network is considered intelligent when it can produce the correct output for new input. The paper by Ivezic and Garret use such methods for design synthesis. They present a neural network system, NETSYN, for estimating the probability of possible attribute values in a given design context. The evolved network represents a subsymbolic abstraction of the relationships between a design's attributes and design specification.

The *evolutionary approach* is based on an analogy to biological genetics. Computational models using the evolutionary approach include genetic algorithms and classifier systems. Genetic algorithms involve the use of a population of solutions represented by their genetic code and a fitness function to search for the best solution. Operations such as crossover and mutation change the current population into the next generation. A genetic code is typically a string that provides the basis for a solution. A fitness function is typically a mathematical function that evaluates whether a particular genetic code should survive to produce a new generation. In this issue, Gero, Louis and Kundu present a genetic algorithm formalism for learning new shape grammars. In their paper, exploration is modeled as an evolutionary learning process where the learning is based on the performance of generalized grammars rather than on specific design cases.

The *inductive approach* has been successfully applied in many domains and applications. The results of induction can be a set of rules, concepts or logical inferences. Induction involves generalizing a set of examples using statistical, probabilistic, or performance measures to produce a selected representation. The result of conceptual clustering is a classification of examples in a representation that can be used to predict the classification of a new example. In their paper, Henderson and Bailin describe how they apply conceptual clustering to dynamically organize a repository of software components for reuse. The results are compared to that of a manual approach and it is argued that a reasonable classification can be generated without human supervision. The paper by Maher and Li shows how conceptual clustering provides a starting point for learning design knowledge, but that it needs to be augmented with additional statistical and probabilistic techniques to learn generalized relationships between attributes within a cluster or "design concept." The result of applying this augmented learning method is a set of design concepts that provide more complete knowledge, in the form of networks of attributes organized into a set of spaces, than the classical conceptual clustering approach.

The *analogical reasoning approach* uses previous solutions or plans to solve a new problem. Various types of analogical reasoning have been developed, for example the transformational approach, in which a previous solution is modified to solve a new problem, and the derivational approach, in which the method used to produce a previous solution is applied to a new problem. Analogical reasoning is also the basis of case-based reasoning. The emphasis in case-based reasoning has been on the organization of case memory and the retrieval of appropriate cases. In this special issue Bhatta and Goel focus on discovery of physical principles by generalizing design experiences. The approach focuses on generalizing over experiences of structure, behavior and function—a symbolic approach in contrast to the sub-symbolic approach of Ivezic and Garret. Bhatta and Goel argue that "learning abstract models . . . that facilitate cross-domain analogical design provides a great potential for machine learning in design because cross-domain analogies often play a crucial role in nonroutine design."

Knowledge compilation methods can use methods from any of the above categories, but have some special characteristics. Knowledge compilation is concerned with producing directly usable knowledge (e.g., rules) from deeper, more fundamental knowledge. In this special issue Chabot and Brown concentrate on compiling knowledge using an incremental learning approach termed Constraint Inheritance. Errors or inadequacies are detected in compiled knowledge and, given an explanation from a deeper form of knowledge, a new constraint can be generated which represents the problem solving domain more efficiently.

This special issue highlights the growing realization that for future computer-based design systems to be more effective they must continually evolve their state of knowledge to reflect new experiences and that they must use that knowledge in all aspects of design problem solving. We thank the authors for their hard work and express our gratitude to the reviewers for their time, effort and valuable comments about the papers. We hope the readers find the papers as interesting and profitable as we did.

REFERENCE

Gero, J.S. (1992). *Artificial Intelligence in Design '92*. Kluwer Academic Publishers, The Netherlands.