A Review of the Fourth International Workshop on Machine Learning

ROGERS P. HALL (RHALL@CIP.UCL.EDU)
Department of Information & Computer Science, University of California, Irvine, CA 92717, U.S.A.

BRIAN FALKENHAINER (FALKEN@A.CS.UIUC.EDU)
Department of Computer Science, University of Illinois at Urbana-Champaign, IL 61801, U.S.A.

NICHOLAS FLANN (FLANN%OREGON-STATE@CSNET-RELAY)
Department of Computer Science, Oregon State University, Corvallis, OR 97331, U.S.A.

STEVE HAMPSON (HAMPSON@ICSE.UCL.EDU)
Department of Information & Computer Science, University of California, Irvine, CA 92717, U.S.A.

ROBERT REINKE (REINKE@UICSL.CSL.UIUC.EDU)
Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, IL 61801, U.S.A.

JEFF SHRAGER (SHRAGER.PA@XEROX.COM)
Xerox Palo Alto Research Center, Palo Alto, CA 94304, U.S.A.

MICHAEL H. SIMS (SIMS@AMES-PLUTO.ARPA)
NASA Ames Research Center, Moffett Field, CA 94035, U.S.A.

PRASAD TADEPALLI (PRASAD@RED.RUTGERS.EDU)
Department of Computer Science, Rutgers University, Piscataway, NJ 08854, U.S.A.

1. Introduction

What happens when you leave 160 machine learning researchers in the Southern California sun for four days? Incrementally examining the condition of each, one might induce that all were well done. Reasoning from a partial theory of social conventions, one might deduce an explanation that they participated in the Fourth International Workshop on Machine Learning, held between June 22 and 25 at the University of California, Irvine. In fact, the latter conclusion is the correct one. The proceedings of the workshop are available through Morgan Kaufmann Publishers, Inc.

Recurring themes. Although many learning issues surfaced during the workshop, discussion (both formal and informal) centered around several themes:

- The division between inductive (empirical) and deductive (explanation-based) learning techniques was still evident. However, there was increasing interest in combining these techniques in a synergistic fashion: using empirical methods to refine inconsistent or incomplete domain theories and using explanation-based methods to focus empirical learning.
- Another theme involved methodologies for verifying claims about machine learning techniques. Approaches to verification ranged from empirical (e.g., Fisher's use of an inference task for incremental conceptual clustering) to analytic (e.g., Haussler's formalization of a simplified version space technique).
- Among researchers working on explanation-based learning, there were several efforts to incorporate previously implicit aspects of context into the machine learning task. For example, Keller's work used a description of performance improvement as a parameter to the learning task.
- There were several unusual approaches to concept representation, each departing from summary descriptions using symbolic logic. For example, Quinlan presented probabilistic extensions to inducing decision trees that accommodate imprecise or missing information. Perhaps a more fundamental departure, Kibler and Aha compared the performance of several exemplar-based learning algorithms which produce purely extensional concept descriptions.

There was a healthy diversity of thought around these central themes, as demonstrated by the variety of papers at the meeting. In the sections that follow, each paper will be briefly reviewed. These reviews are organized to broadly reflect the tasks that researchers address, and within these tasks, the particular techniques adopted. A set of more general papers addressing cognitive architectures and formalizations of learning techniques is reviewed last.

2. Learning and classification

Exemplar-based approaches. Rather than constructing only abstract descriptions, exemplar-based learning techniques also store exemplars of each concept that is to be learned. When a new instance must be classified, it is compared to stored exemplars and assigned to the category containing the best-matched exemplar. A major advantage of this approach is to allow a simple representation for exceptions and disjunctions. Exemplar-based learning systems differ in how they decide which exemplars to store, how

they index the stored exemplars, and in how they compute the best match to a new instance.

Bradshaw points out that a viable alternative to the traditional search for concept descriptions in an hypothesis space is to “seed” instances into that space and leave “generalization” to the matching algorithm. He describes NEXUS, a speech recognition system that incorporates instance-based learning methods. When a new instance is classified by the system, two forms of learning are possible. If the initial classification was correct, NEXUS combines the reference pattern with the training example using a weighted averaging scheme. When the initial classification is incorrect, NEXUS seeds the instance into the concept space under the assumption that it is significantly different from existing concepts. After learning, the system’s error rate on a speech recognition task is only one-seventh that of a system based on traditional template-matching.

Bareiss, Porter and Wier present work on PROTOS, an exemplar-based learning apprentice system for diagnosis in clinical audiology. The system acquires exemplars and domain-specific knowledge for a heuristic classification task by serving an apprenticeship with an expert teacher. Given a new instance to be classified, PROTOS retrieves exemplars from memory using three indexing strategies: features extracted from the instance are used as indices into general categories and exemplars; exemplars within categories are indexed (ordered) by prototypicality; and exemplars are indexed to category neighbors by differences. Knowledge encoded in these indexing schemes is modified as the system solves new problems. PROTOS follows a strategy of “lazy generalization,” creating more abstract descriptions of exemplars when all differences between an instance and exemplar can be explained by existing knowledge (and the teacher concurs). Much like an explanation-based learning system, PROTOS makes only those generalizations implied by its domain knowledge. Unlike such systems, it acquires initial domain knowledge through explanations provided by the teacher when the system fails to make a correct classification. The paper presents a sample diagnostic session with PROTOS, and during their presentation the authors showed that the system improves its classification ability as it is exposed to new cases.

Kibler and Aha describe experiments that test the performance and storage efficiency of three simple methods for deciding which exemplars to store as part of a concept definition. The “proximity” method simply stores all exemplars. The “growth” method stores only those exemplars that are incorrectly classified when first seen. The “shrink” method finds a *correct* subset of all the exemplars by removing those that would be classified correctly by remaining exemplars. Each method is applied to Quinlan’s data on thyroid patients. Results of the experiments suggest that the “growth” and “shrink” algorithms are about equally efficient and accurate, that both

are much more efficient than the proximity method, and that both are only slightly less accurate than the proximity method.

Probabilistic approaches. Traditional approaches to learning logical concept descriptions require finding conditions that define class membership and then classifying an instance precisely within some conceptual class. However, imprecise classification may be desirable or even necessary in some practical settings, e.g., where instances are incompletely or incorrectly described. In the following set of papers, probabilistic methods are used to determine the relative contribution of conditions for defining concept membership or to decide whether a classified instance is a typical member of some conceptual class.

Quinlan refines his earlier methods for decision tree induction to support uncertain classification. Decision trees typically assign examples to a single class based on precisely matched values over a set of features. Quinlan uses performance statistics for an induced decision tree to reduce its size and to allow probabilistic classification of new instances. These measures are implemented in C4, the successor to Quinlan's (1986) widely-used ID3 program. The C4 system produces a decision tree that assigns class membership probabilities to new instances, allowing uncertainty in classification, and "softens" the branching boundaries in the tree, avoiding changes in classification caused by minor differences in an attribute value. In his talk, Quinlan presented results on thyroid disease diagnosis that suggested C4 can produce better decision trees than ID3.

Michalski extends the notion of imprecise classification to include concepts whose meanings vary with context. His solution is a two-tiered representation consisting of a Base Concept Representation and an Inferential Concept Interpretation. The base representation is a declarative statement "describing a general and easy-to-define meaning of the concept and its typical purpose or use." The interpretation is a context-sensitive inferential process that uses domain knowledge and multiple inference strategies to determine whether a given object matches the base representation in a particular situation. A simplified version of this scheme is tested in a medical diagnosis domain (lymphography) using Michalski et al.'s AQ15 (1986) to construct concept descriptions that correctly classify observed training examples. Statistics on the relevance of each part of the concept description are collected during learning and used to eliminate marginal components from the base representation. A simple heuristic interpretation that punishes mismatch on highly typical features and prefers frequently observed structures is used to test the derived base representations on new cases. Results suggest that a smaller concept representation and a slightly more complex interpretation perform as accurately as full concept descriptions applied with strict matching.

Fisher describes COBWEB, an incremental conceptual clustering system that uses probabilistic partitioning heuristic to cluster instances into a hierarchically organized set of partitions. The heuristic improves the system's ability to predict unknown features given knowledge of category membership, precisely the inference problem that motivates studies of classification. While existing conceptual clustering systems help to summarize data in an understandable manner, evaluation of their performance is usually subjective. Fisher adopts a novel approach to evaluation in which classification by COBWEB's conceptual taxonomy is compared with classification by traditional methods that learn from examples. Fisher uses a reconstruction of Quinlan's (1986) ID3 system to build a decision tree for each measured attribute. To compare learned concepts in COBWEB and ID3, each attribute is treated as a target (unknown) concept in a classification task. Tested over individual attributes, classification accuracy with COBWEB's concept hierarchy approximates the predictive capacity of individual decision trees generated using ID3.

Wilson describes BOOLE, a system that uses a "genetic algorithm" to learn disjunctive boolean concepts. The system searches a theory space by selecting among "parent" concepts according to their strength. Strength is assigned to concepts by a reinforcement component that rewards or punishes their performance in interpreting the environment. Selected concepts are altered by "crossover" or "mutation" operators to form new "offspring" concepts. These offspring replace the weakest existing concepts in the population and then compete for reinforcement in subsequent performance trials. Wilson shows that this method can learn a difficult boolean function (the 6-multiplexer) to 91% accuracy in only 3200 trials, and to 97% accuracy in 12,000 trials. Although the genetic learning approach is briefly contrasted with other computational learning techniques, success on the multiplexer concept is not compared across techniques.

Concept learning and bias. Some problems in machine learning appear so difficult that relatively little work has yet been done. The problem of *constructive induction* (Dietterich & Michalski, 1983) appears to be one of these problems. Simply stated the problem is one of dynamically shifting the representation language (or bias) to improve learning performance, either by selecting or generating new representational terms. Two papers make interesting contributions in this area.

Rendell, Seshu and Tchong describe VBMS, a variable bias management system that dynamically selects its representation language and learning algorithm. The system monitors its progress on induction tasks and selects biases based on characteristics of the particular induction problems presented. These are defined by the user and include the number of training instances and the number of features. VBMS learns to associate problems

with appropriate biases by using a clustering algorithm (PLS1) to partition the space of induction problems. Resulting classes of problems have similar relative effectiveness using various biases and learning algorithms.

Schlimmer describes a triad of learning methods that work together to minimize the impact of initial representation decisions in the concept acquisition system, STAGGER. A connectionist style learning method modifies a simple concept description by changing weights that represent the sufficiency and necessity of features. A second, symbolic learning method forms boolean combinations of these descriptive elements and allows the first method to overcome its representational shortcomings. The third, numerical learning method partitions real-valued attributes into discretely-valued ones, thereby allowing the other two methods to construct more effective concept descriptions. In effect, the boolean method relaxes the bias of the weight adjusting method, and the numerical method changes the bias for both the weight and boolean methods. Schlimmer contrasts learning with and without bias adjustment.

3. Learning in problem solving and planning

Heuristic search approaches. A central metaphor in AI and machine learning is viewing intelligence as search through a space of representational descriptions. Papers in this section focus on improving task performance by learning heuristics to control this search. In the last paper, Mostow casts the problem more abstractly by asking how to control a learner's search for effective search heuristics.

Keller describes METALEX, a system that improves the performance of a symbolic integration problem solver under different specifications of the target concept and the capabilities of the problem solver. This knowledge of the learning context has been encoded in earlier systems through fixed, implicit assumptions made at design time. In contrast, METALEX takes a specification of the learning context as an explicit input, making the system more amenable to change. Inputs to METALEX include a non-operational description of the target concept, a domain theory, a description of the performance system, and a set of training problems. In addition performance objectives are explicitly given in descriptions of desired efficiency (time taken to solve the training problems) and effectiveness (percent of training problems correctly solved). The system constructs an operational description of the target concept that satisfies the performance objectives. Keller reports experiments with the system's success given various combinations of training problems and performance objectives.

C. W. Anderson shows how a multilayer network learning system can be applied to an exemplary control problem, the pole-balancing task. An error back-propagation algorithm is used to train the multilayer networks

used in this problem. Two learning issues are addressed: acquiring arbitrary stimulus-response mappings and achieving real-time credit assignment even though failure (dropping the pole) may be signalled after an arbitrary number of actions. Anderson describes an adaptive critic that learns to give immediate evaluation of system actions. Results show that a two-layer system dramatically outperforms a single-layer network after first-layer units have acquired “helpful” features (e.g., a numeric representation of the pole falling to the right).

Utgoff and Saxena investigate inductive techniques for improving best-first search. Although heuristic evaluation functions are usually treated numerically, they can be more generally viewed as a two-place *preference predicate* over states. A Best First Preference (BFP) search method uses Quinlan’s (1986) ID3 program to induce this preference predicate. Initially, the system finds the best solution path for a problem. For each node on that path, every alternative (non-best) node in the frontier is used to give (best-node, non-best-node) pairs as positive examples of *preferred* and (non-best-node, best-node) as negative examples. BFP is trained on the 8-puzzle task, using best solution paths for randomly generated training problems. The system learns a preference predicate that solves nine out of ten randomly generated (but unseen) test problems after training on 50 solved problems.

Mostow compares three systems that perform operationalization. One is Mostow’s own system, BAR, which operationalizes advice in the card game of Hearts. The other two are Keller’s METALEX, which works in the domain of integration problem solving, and Kedar-Cabelli’s Prolog implementation of EBG. Mostow casts these systems as searching a space of concept descriptions with the goal of finding an operational description of the given target concept. For each system, he identifies several forms of search control knowledge, their sources in the systems, and how they are used. For example, both METALEX and BAR employ an “operationality criterion” to determine when the search is to terminate, while EBG stops searching once the explanation is complete. One important observation from this analysis is that none of these approaches fully exploit all forms of control knowledge.

Planning approaches. Beyond problems involved in learning individual operators are problems of learning plans or sequences of operators that can be used to achieve a complex set of system goals. Difficulties with plan recognition, credit assignment, and goal interaction complicate the general problem of improving performance in some task domain.

Minton, Carbonell, Etzioni, Knoblock, and Kuokka describe PRODIGY, a general-purpose planning architecture that uses explanation-based learning to acquire control knowledge during problem solving. The authors

consider how to ensure that the system learns only *effective* control knowledge, where the benefits of applying the learned knowledge must outweigh the costs of testing to see if the knowledge is applicable. PRODIGY tackles this problem in three ways. First, varied explanations are constructed that allow learning from a variety of planning situations including success, failure, and goal interaction. Second, the resulting explanation is simplified to reduce its match cost through partial evaluation and the application of domain knowledge. Finally, the utility of the resulting control rule is monitored during subsequent problem solving by keeping track of the cumulative cost of matching versus the cumulative savings in search time. PRODIGY has been applied in a variety of domains including the blocks world, machine shop scheduling and three dimensional robotics.

McCluskey describes FM, a problem solving system that uses chunking. Given a successful solution to a problem (e.g., a plan for robot movement), FM generates a chunk for each operator in the solution. These chunks function as search control heuristics and consist of the operator, the goal, and the weakest preconditions to that point in the sequence. Later FM uses these chunks in a forward state-space search to help select an operator for the next search step. FM also learns chunks which are used to focus the instantiation of operators or macros in goal reduction search.

Hammond describes a case-based approach to constructing explanations for planning failures and repairing faulty plans. The idea is to search within a memory of explanations for similar, previously experienced failures in the hopes of avoiding similar problems in a current case. He proposes a small vocabulary of failure types that are used as indices into case memory. These include priority conflicts for known goals, goal/plan inconsistency, atypical plan instantiation (e.g., unexpected role fillers), and unexpected outcomes (e.g., a mechanism fails). Each failure type is used as a top-level node in a discrimination network of prior cases. With each failure type, Hammond stores explanation strategies, suggestions for indexing new cases, and a set of discriminating indices to prior cases. In the best case, a retrieved explanation will fit the current situation perfectly; in the worst case, the retrieved explanation must be rejected and a new explanation constructed "from scratch." Hammond argues that this approach works by explicitly chunking the space of possible explanations into "islands of viable explanations" reflecting the learner's actual experience.

O'Rorke reports empirical tests of the validity of explanation-based learning mechanisms. A reconstruction of Newell, Shaw and Simon's (1963) LT, an early theorem prover for propositional calculus, is augmented with the capacity for explanation-based learning. On 52 problems (theorems) from Whitehead and Russell's *Principia Mathematica*, O'Rorke applies LT under three conditions: without any learning, with rote learning of theorems, and with explanation-based methods that generalize successfully proved theo-

rems. The non-learning version of LT solves 16 problems, rote-learning LT solves 41 problems, and the explanation-based version solves 43 of the 52 problems. In eight instances, the third system produced theorems that were strictly more general than the one originally presented in *Principia Mathematica*. This is noteworthy in that generality was clearly a major goal in original construction of the theorems.

Kodratoff and Tecuci describe DISCIPLE, a learning apprentice system that uses explanations to learn a generalization of a user-given rule. In explanation-based learning, an explanation is usually a *connected* tree of inferences which *completely* explains an instance. However, in DISCIPLE an explanation is simply a *set* of justifying inferences which might be neither connected nor complete. Given an instantiated rule, the system generalizes this rule, generates a plausible set of justifications for the rule, asks the user to verify that each of these justifications is important for the explanation, and then uses this explanation to build necessary and sufficient conditions for the rule. By cleverly interrogating the user, DISCIPLE can utilize a weak or incomplete domain theory that is not sufficient to generate an explanation in the usual sense.

Kolodner describes JULIA, a system that plans catered meals using case-based problem solving. Given a catering task, retrieved meal planning cases are used to avoid similar failures, to reuse portions of previously successful meal plans, and to build generalized schemata from related cases. JULIA consists of case-based and problem-reduction problem solvers coordinated within a blackboard architecture. Goals posted on the blackboard during problem reduction serve as retrieval probes for generalized or case-specific knowledge residing in a discrimination-based memory of prior problem solving episodes. The case-based problem solver focuses on current task goals if a successful case is retrieved, but when retrieving a case of failure, it shifts focus to avoid that failure in the current problem. The paper includes a meal-planning example in which prior cases of problem solving success and failure are retrieved, constraints are transferred, failures are anticipated, and an inconsistent analogy is abandoned.

4. Learning and natural language

One of the most fundamental aspects of human development is language acquisition. Machine learning research in this area addresses a number of central problems: accounting for the acquisition of syntactic and semantic knowledge, separating language specific forms of learning from more general forms, and learning to find unambiguous interpretations as a general performance goal. The papers in this section address each of these issues.

Lehnert describes ELAN, a sentence analyzer that learns to parse sentences into case frame representations when trained on correctly parsed ex-

amples. The system begins with a lexical dictionary, knowledge of syntax encoded in a simple chart parser, and knowledge of semantics represented as case frames. During training, the system acquires “integration maps” between syntactic (e.g., noun phrase) and semantic (e.g., actor) elements of a given sentence. Given a novel sentence during testing, learned mappings are assembled into a constraint network with inhibition links between competing interpretations. From an initial activation based on previous usage, a relaxation algorithm searches for a stable pathway in the constraint network. If found, this pathway gives a consistent surface interpretation of the novel sentence. As a last step, ELAN further instantiates a frame representation of this interpretation.

Zernik describes RINA, a language learner that forms a hierarchy of lexical phrases structured by generality. Given a sentence and sufficient information to determine the appropriate interpretation, previously unknown idiomatic phrases (e.g., *stashed it away*) are incorporated into the hierarchical lexicon by empirical methods or explanations provided by metaphors the system already understands. In either case, future performance is improved. RINA is offered as evidence that general learning techniques are appropriate for language acquisition.

Martin approaches the problem of how to extend a lexicon of word senses to handle newly encountered, metaphorical uses of known words. Learning mechanisms are tested as part of a conversational UNIX consultant system which learns to interpret questions like, “How can I *get out* of lisp?” Three approaches to acquiring one-to-many mappings between words and meanings are examined, each representative of a small set of regular similarities that provide structure to the lexicon. For example, “metaphorical frame semantics” uses the principle that words within a single semantic frame (e.g., knowledge of containers) have known relationships with one another. When a member of the frame is found in a novel utterance (e.g., *get out* of lisp), the new meaning can be understood by finding another member of the frame (e.g., *get into*) that engages the same object (i.e., lisp) via a known metaphor. In this case, the known metaphor maps *get into* to a meaning of enablement in the domain of lisp programming. Since *get out* and *get into* are known to participate in a reversible-state-change relation for containers, that same relation can be applied in the lisp domain to acquire the meaning of “disable.” All three approaches use a form of intersection search to find pathways between lexical items in the existing lexicon, and then update the lexicon to reflect successful analogical comprehension.

5. Machine discovery

Machine discovery covers a broad spectrum ranging from empirical discovery, that derives general laws describing observed data, to theory for-

mation, that goes beyond description to find explanations for these laws or observations. Most work has centered on empirical discovery, using simple algorithms in knowledge-poor environments. However, discovery often combines a number of reasoning and learning techniques. More recent work recognizes this complexity by integrating knowledge-intensive learning methods with more ambitious reasoning systems. The following set of papers apply inductive, deductive, and analogical learning techniques, together with a more active account of observation and experimentation, to a variety of discovery tasks.

Observational discovery. These papers address the task of discovery when the system is restricted to a relatively passive role with respect to environmental observations.

VanLehn and Garlick describe an automatic protocol analysis tool called CIRRUS. From a machine learning perspective, this could be viewed as a model of scientific discovery in the domain of cognitive psychology. The system parses manually-encoded protocols of children's subtraction behaviors according to a grammar that reflects a theory of subtraction. The resulting parse is augmented with hypothetical internal states of the student, assuming the student's activities resemble an agenda-based problem solver following an existing plan. Finally, agenda selections from the augmented parse are passed to a version of Quinlan's (1986) ID3 algorithm. The resulting decision tree provides an explanatory and predictive model of the student's subtraction behavior. Protocols which CIRRUS is unable to model signal opportunities for revising or extending the underlying theory of subtraction. As an experiment in theory validation, CIRRUS is an interesting and important contribution to machine learning. The utility of CIRRUS as an automatic protocol analysis tool is being explored.

Falkenhainer describes PHINEAS, a system that uses verification-based analogical reasoning to discover qualitative theories about physical processes. This work addresses open problems in qualitative reasoning and discovery: existing qualitative reasoning systems operate with fixed domain theories, and existing discovery systems do not find explanatory models or attempt to confirm discovered material. Qualitative Process Theory (Forbus, 1984) provides process-level explanations for physical observation (behavioral histories), and theory-driven envisionment provides a method for verifying analogically-derived additions to the process theory. An example is shown in which the system is unable to explain an observation of temperature equilibrium when objects with different initial temperatures are placed in contact. Given a previously experienced water flow history that shows similar behavior, PHINEAS attempts to use a process-level explanation of the water flow history to explain temperature equilibrium. If the water flow process description can be transferred and used to construct

an envisionment covering the observed temperature history, the process theory has been successfully extended. Otherwise, Falkenhainer proposes theory refinement strategies such as experimentation or attempting multiple analogies.

Pazzani explores the acquisition of causal relationships in the absence of a causal theory. He draws on developmental data to argue that children use very simple rules to induce cause where they observe "intra-example relationships." For instance, if a child sees someone blow air into a balloon and then sees that the balloon is inflated, she might assume a causal connection between these events. Pazzani's describes OCCAM, a system that uses these rules when it cannot find a theory to explain observations. Causes induced by these rules can be subsequently used to formulate a theory for the domain. In motivation and scope, Pazzani's work is quite similar to Anderson's use of causal induction in the PUPS framework (reviewed below) for "boot-strapping" causal theories in a naive learner.

Mozetic examines the problem of constructing deep models of domains in which pre-existing models are not available. Both data-driven inductive inference and model-driven debugging (Shapiro, 1983) are used to construct a qualitative model of the heart. Given a basic component model of the heart and a few instances of its behavior, the system inductively hypothesizes functions of components, thus generating an initial partial model. The model is then actively refined through a cycle of prediction generation and confirmation. One of the key contributions of the work lies in the use of a hierarchical representation and in the development of a model from abstract levels to successively greater levels of detail. Starting from abstract models simplifies the reasoning and inductive learning tasks. Furthermore, it enables the model debugger to use previously learned abstract models in guiding the search for corrections to incomplete hypotheses.

Discovery and experimentation. Moving beyond passive observation, another set of papers viewed discovery as an active process of hypothesis generation and confirmation. Experimental interventions within the environment are planned, conducted and evaluated as critical tests of theoretical accounts of environmental contingencies.

Carbonell and Gil present a system that improves its planning performance by learning a more accurate model of the environment through failure-driven experimentation. The system is intended to be a general purpose planning system (part of Minton et al.'s PRODIGY, discussed above) and is illustrated in the domain of telescope mirror production. Given an incomplete domain theory and a planning goal, the system proceeds to construct a plan that, according to the current domain theory, will achieve the goal. Following each operator application, the state of the external environment is observed to determine if the plan is executing as predicted.

The authors identify three failure conditions, each of which illustrates an error in the specification of an operator. For example, if an operator is to be applied and at least one of its preconditions is unexpectedly unsatisfied in the environment, then one of the previously applied operators is deemed to have changed the missing precondition. An experiment strategy is formulated that performs binary search on this sequence to determine the offending operator. Once identified, the negation of the missing condition is added to the operator's post-condition. In general, the experimental system is capable of finding missing pre- and post-conditions on operators and learning control rules that stem from unanticipated goal interactions.

Sammur and Hume describe how observations, current theories, and experiments interplay in the process of theory refinement in a reactive environment. They describe CAP, a system in which a "child" robot learns by imitating the observed behavior of a "parent" robot interacting with a simulated world. Thus learning occurs without direct supervision. They present an example in which the learner refines its conception about pouring liquids into containers. The child watches the parent take actions that alter the world state and then stores these observations as state sequences. Later, the child imitates parent behavior by finding a maximal partial match between a current state and some state in a stored sequence. The child uses the partial match to generalize subsequent states in the stored sequence, infers a series of actions that should yield the generalized sequence, and then executes those actions in the simulated world. Monitoring the outcome of such experimental plans in the simulated world lets the learner refine its theory of that world by modifying enablement conditions on actions. Sammur and Hume point out the difficulty of guiding and constraining such a learning system. For example, they constrain the search for partial matches by using only information that changes from one state to another and eliminating any material that describes robot actions.

Sims investigates interactions between reasoning techniques in the discovery process (e.g., induction from examples, experimentation, and formal proofs based on current models). He describes IL, a discovery system that provides a flexible architecture for integrating multiple reasoning techniques. The system explores mathematical domains in a manner similar to AM (Lenat, 1977), using a production system that is organized as an agenda of tasks ordered by a dynamic estimation of worth. Rules may add new tasks, add new knowledge, call upon explanation-based learning, or invoke example verification. Sims describes the results of IL in the area of mathematics called Conway numbers. When proving that zero is the additive identity, the system guides the proof of the general case by applying explanation-based generalization to the proof of a specific example.

Zytkow concurs with Sims' basic theme, treating scientific discovery as the complex interaction among a variety of search techniques. He describes

FAHRENHEIT, a system for inducing functional formulas from observation that is built on top of BACON.3 (Langley, 1981). A number of search methods are cleanly integrated through the use of a search interpreter that controls a stack of search templates. This architecture simplifies the conceptual complexity of the system and enables the exploration of more extensive uses of search in discovery. For example, Zytkow's system determines the scope of a detected law by experimentally seeking its limit points and applying BACON-like methods recursively to the uncovered boundary data, thus providing equations that describe scope boundaries.

6. Cognitive architectures for learning

Finding simple but very general learning mechanisms continues to be the "holy grail" of many machine learning researchers, particularly those interested in psychological explanation. The usual approach is to embed a plausible learning mechanism within a general reasoning architecture and then to show that these arrangements support reasoning and learning across a wide variety of task domains. The papers in this section describe refinements or further task variety for two widely-studied projects and a new proposal for resource-limited learning.

J. R. Anderson describes the PUPS theory of learning, using the domain of tutored algebraic problem solving as an example. The learning problem is to build an explanatory domain theory and to improve problem solving performance through interaction with the tutor on problems like "Isolate X in $3X = 6$." In contrast to his earlier ACT* framework, in which knowledge compilation provided a deductive learning mechanism (Anderson, 1983), the PUPS theory adds *causal induction* and *analogical extrapolation* to generate new declarative structures that can be compiled into rules. Thus the PUPS learner both induces a domain theory from pre-theoretic observations and improves its problem solving performance through deductive compilation. Causal induction (e.g., that typing $(+ 2 3)$ causes a LISP machine to respond with 5) depends upon the spatio-temporal contiguity, similarity, and statistical regularity of cause and effect. "Safe" analogical extrapolation (e.g., predicting the result of $(+ 3 5)$ by analogy to $(+ 2 3)$) assumes that if compared entities are from the same category or play a similar functional role then they will have similar causal properties. As implemented, analogical learning may not be safe and proceeds despite violations in category membership or functional preconditions. Anderson presents an extended example in which a PUPS learner induces causal relations between tutor-suggested steps in a solution to the variable isolation problem shown above. These causal links could be extrapolated by analogy if the learner were subsequently given a similar algebra problem.

The paper by Steier, Laird, Newell, Rosenbloom, Flynn, Golding, Polk, Shivers, Unruh, and Yost receives the workshop's "most authors award" by presenting a collection of abstracts that demonstrate the breadth of SOAR's learning accomplishments during the past year. Historically, the SOAR group has shown that many sorts of tasks can be expressed as problem solving. This is a standard theme in AI, but they have also shown that adding a type of back-tracing memory system (chunking) to the problem-solver enables useful learning. The eight projects abstracted in this summary paper demonstrate a great deal of procedural learning and begin to hint at declarative learning with studies of task acquisition and "data chunking." Incorporating a robust declarative learning capacity within SOAR would be a significant extension to the architecture, and these efforts provide a start towards this goal. The authors conclude that "the diversity of problem-solving of which SOAR is capable is directly responsible for the varieties of learning in SOAR." The extent of what is contributed by SOAR itself, versus "SOARware engineering," is becoming more clear.

Langley, Gennari and Iba distinguish between "search-based" and "hill-climbing" models. A search-based model considers multiple alternative hypotheses or is able to backtrack out of bad decisions. A hill-climbing model considers only one hypothesis at a time and always moves forward to a new theory. The authors argue that the hill-climbing paradigm is both more space efficient and more psychologically consistent for domains of perceptual and skilled activity. In these domains, humans appear to act with a single perceptual or motor schema and to modify it according to new inputs. They do not seem to consider alternative schemata (at least not explicitly) and do not seem to store and reprocess old stimuli when they learn. The authors introduce CLASSIT and MAGGIE as examples of the hill-climbing paradigm. CLASSIT is an incremental conceptual clustering system that forms a hierarchy of physical object concepts, while MAGGIE refines motor skills through practice. They also show how a number of historical models (including Winston's thesis and explanation-based models) fall into the hill-climbing metaphor.

7. Formalizations of machine learning techniques

One problem with comparing various learning techniques is finding a meaningful baseline model for comparison. Each paper in this section evaluates known learning techniques against some kind of formal model. The first two papers present theoretical results in Valiant's (1984) *learnability* framework. In this framework, an algorithm is said to learn from examples if, in polynomial time and with high likelihood, it can find a rule that is reasonably accurate. This means that the rule found by the algorithm will be able to predict future events drawn from the same distribution on which

it has learned and make these predictions with controllable error.

Haussler shows that Mitchell's (1982) version-space algorithm for concept learning has worst case time complexity that is exponential in the number of examples, even for simple classes of hypothesis spaces like purely conjunctive concepts. He then gives a "one-sided" learning algorithm for purely conjunctive concepts that runs in polynomial time with the number of examples. One of the main contributions of Haussler's paper is a measure of expressiveness of a hypothesis space that can be used to derive an upper bound on the number of examples needed for convergence in Valiant's learnability model. For example, using this measure, the classes of k -CNF (arbitrary conjunctions of pure disjunctive concepts, each disjunction using at most k atoms) and k -DNF (similar, but for disjunctions) concepts are shown to be learnable in polynomial time for any fixed k .

Kearns, Li, Pitt, and Valiant survey results within Valiant's learnability model, introduce several theoretical tools that can be used in proving learnability results, and suggest some directions in which the model can be extended. The authors extend the learnability model as follows: class A is learnable by class B *iff* all the conditions of learnability are satisfied except that the output formula is from class B . Interestingly, k -term-DNF (disjunctions of at most k pure conjunctive concepts) is *not* learnable by k -term-DNF itself (for $k > 1$), but it *is* learnable by k -CNF for any fixed k . This shows that concepts can sometimes be made learnable by representing them in a richer language! Further extensions to the learnability model include allowing noise in the data, allowing the learner to ask questions, and restricting the probability distributions on the examples. A tantalizing open problem is whether efficient algorithms can be found when k is large.

Rivest and Schapire present a new algorithm to learn the structure of a deterministic finite state environment by experimentation. The learner has no *a priori* knowledge other than a legal set of actions and a set of possible sensations. A *test* is a sequence of actions followed by a predicted sensation. Two tests are equivalent if they have the same value at any global state. Their algorithm works by systematically experimenting with the environment to produce an equivalence relation on tests. A key insight of the paper is that for many "natural" environments, the number of the equivalence classes of tests (termed *diversity*) is much smaller than the number of states of the environment. Their algorithm is polynomial in the diversity rather than the size of the environment. For example, the structure of the Rubik's Cube (over 10^{19} states) was inferred in less than seven minutes of CPU time. They conclude by proposing several extensions to their model, including actions with probabilistic or conditional effects.

Kedar-Cabelli and McCarty give a Prolog implementation that performs explanation-based learning as a by-product of resolution theorem proving. The implementation is derived from the pure Prolog meta-interpreter

(Sterling & Shapiro, 1986) through a series of simple extensions and requires fewer than 12 lines of code. Given an example of the target concept, the system constructs both the specific proof tree for the example and the generalized proof tree representing the explanation. The paper demonstrates the system learning a generalized concept of suicide from a single example. The inputs are the target concept (a Prolog goal), the domain theory (a set of Prolog clauses), and the training instance (a set of Prolog facts). A post-processing stage is employed to operationalize the resulting explanation by extracting a conjunction of known literals from the generalized proof tree. In addition to presenting a Prolog implementation, the authors correct an error in an earlier EBG algorithm (Mitchell, Keller, & Kedar-Cabelli, 1986).

Russell identifies three sources of validity for analogical inference and relates these to the problem of generalizing from a single instance. The first draws its validity from the presence of an analogical hint provided by a tutor (e.g., 'A is like B') which guarantees that certain aspects of B actually hold for A. A second type of valid analogical inference occurs when the domain theory is strong enough to derive the inference in the absence of a particular example. Russell's contribution lies in identifying a third source of valid inference, *determinations*, that are reminiscent of functional dependencies. P is said to determine Q when particular values for the set of functions P correspond to unique value assignments for functions Q (e.g., nationality determines native language). This relation enables valid analogical inference as well as valid single-instance generalization, yet requires both the rule and an instance. Thus, determinations represent a slightly weaker form of domain theory that is still useful for explanation-based generalization when a stronger theory is unavailable.

Stepp describes two systems, INDUCE/NE and CLUSTER/CA, which provide partial solutions to problems encountered when inducing concepts over structured objects. He emphasizes the importance of clarifying whether systems learn concepts for what an example *contains* versus what an example *is*. In the latter case, the concept can lead to undesirable results under interpretation. For example, a tiny arch on top of a huge building clearly contains the appropriate parts, but the building itself should not be classified as an arch. Stepp incorporates boundary (must, must-not) conditions into concept descriptions induced by INDUCE/NE to partially overcome this problem. Another difficulty with structured objects is that their representational diversity admits to an immense space of possible concept descriptions. CLUSTER/CA is a conceptual clustering system which uses background knowledge about the learning purpose and environment to help select relevant representational terms during induction.

References

- Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- Dietterich, T. G., & Michalski, R. S. (1983). A comparative review of selected methods for learning from examples. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach*. Los Altos, CA: Morgan Kaufmann.
- Forbus, K. D. (1984). Qualitative process theory. *Artificial Intelligence*, 24, 85–168.
- Langley, P. (1981). Data-driven discovery of physical laws. *Cognitive Science*, 5, 31–54.
- Lenat, D. B. (1977). Automated theory formation in mathematics. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (pp. 833–842). Cambridge, MA: Morgan Kaufmann.
- Michalski, R. S. (1983). A theory and methodology of inductive learning. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach*. Los Altos, CA: Morgan Kaufmann.
- Michalski, R. S., Mozetic, I., Hong, J., & Lavrac, N. (1986). The multi-purpose incremental learning system AQ15 and its testing application to medical domains. *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp. 1041–1045). Philadelphia, PA: Morgan Kaufmann.
- Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, 18, 203–226.
- Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. *Machine Learning*, 1, 47–80.
- Newell, A., Shaw, J. C., & Simon, H. A. (1963). Empirical explorations with the logic theory machine: A case study in heuristics. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought*. New York: McGraw-Hill.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.
- Shapiro, E. Y. (1983). *Algorithmic program debugging*. Cambridge, MA: MIT Press.
- Sterling, L., & Shapiro, E. (1986). *The art of Prolog: Advanced programming techniques*. Cambridge, MA: MIT Press.
- Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27, 1134–1142.