

A Review of Machine Learning at AAI-87

RUSSELL GREINER (GREINER@AI.TORONTO.EDU)
*Department of Computer Science, University of Toronto,
Toronto, Ontario M5S 1A4, CANADA*

BERNARD SILVER (SILVER@AI.AI.MIT.EDU)
*Computer and Intelligent Systems Laboratory, GTE Laboratories Incorporated,
40 Sylvan Road, Waltham MA 02254, U.S.A*

SUE BECKER (BECKER@AI.TORONTO.EDU)
MICHAEL GRÜNINGER (GRUNINGER@AI.TORONTO.EDU)
*Department of Computer Science, University of Toronto,
Toronto, Ontario M5S 1A4, CANADA*

1. Introduction

Some of us can remember the first AAI conference in 1980 - a cozy gathering of 400 AI researchers tucked away in one corner of a university campus. There were only two parallel sessions of papers, and they filled one relatively thin proceedings volume. There were no tutorial sessions and no exhibition hall. For better or worse, the field of artificial intelligence has grown considerably in the subsequent seven years. The recent Sixth National Conference on Artificial Intelligence, AAI-87, involved over 6,700 people and required the full accommodations of the Seattle Center in Seattle, Washington (site of a former world's fair) for its four parallel technical sessions and exhibition show; and it still needed the University of Washington campus for its four parallel tutorial sessions.

Machine learning (ML) has also developed over these years. It has recently emerged as the subfield of AI that deals with techniques for improving the performance of a computational system. It is now distinguished from studies of human learning and from specific knowledge acquisition (KA) tools. In addition to several ML paper sessions, there are now tutorials on the topic and even ML programs and books on display in the trade show.

This note focuses on the ML-related articles and surveys presented in this AAI-87 conference. The next four sections cover the relevant presentations, including all ten ML and three KA papers and the directly relevant surveys.¹

¹These papers were selected from the 68 ML and 28 KA papers submitted to the conference. The conference proceedings are published by Morgan Kaufmann Publishers, Inc. (ISBN 0-934613-42-7). Although the surveys were not included in the proceedings, Morgan Kaufmann plans to publish an edited version of the transcribed presentations. *Canadian Artificial Intelligence* (October, 1987) includes a more general review of this conference.

Section 2 deals with particular classification techniques, including three technical papers and one survey talk. Section 3 describes the four papers that focus on “knowledge-rich” learning systems, including various explanation-based learning (EBL) systems. Section 4 discusses three papers that present theoretical results. Section 5 describes three papers in the related area of knowledge acquisition. Section 6 covers, in broad brushstrokes, some of the non-ML presentations considered relevant to the ML community, mentioning some papers on analogy, theory revision, and other topics. The concluding section discusses the perceived trends in our field and raises some general issues.

2. Classification techniques

Until recently, “learning” referred almost exclusively to classification mechanisms, focusing on programs that learn concept descriptions from a series of examples and counterexamples. While learning now extends to include many other topics and types of systems, classification is still a very active subfield. Quinlan presented an excellent summary of these classification techniques in his survey talk, “Data-Driven Approaches to Learning Classification Rules.” These systems have only a minimal domain theory and rely almost exclusively on the training examples to learn an appropriate classification function. The purpose of the classification function is to map sets of feature values onto meaningful (but not necessarily pre-defined) classes. He explained some of the difficulties of this task and presented many of the dimensions along which systems can differ – for example, the complexity of the input language, whether or not the data are given incrementally, and so on. He used these dimensions to compare a thorough cross-section of systems, including Michalski et al.’s AQ family of algorithms, his own ID3 system, Breiman’s greedy approach, Stanfill and Waltz’s exemplar-based approach, Schlimmer and Granger’s STAGGER system, and the connectionist approach to learning classifications. He also mentioned some theoretical results in the area that point to polynomial learnability of decision lists.

Two technical papers provide theoretical results on induction techniques; these are discussed in Section 4. The rest of this section discusses the three papers that describe particular ways of finding an apt classification function.

Connell and Utgoff’s “Learning to Control a Dynamic Physical System” describes how their CART program learns to balance an inverted pole on a cart. At each time step, the performance system² pushes the cart (with a fixed force) to the left or to the right, based on the current cart-pole state, which includes the position and velocity of the cart and the angular position and angular velocity of its pole.

To decide which direction to push, CART first estimates the states resulting from each of the two actions and then approximates each state’s “desirability” using an interpolation function over the “desirability labels” of a few (previously memorized) states. CART pushes the cart in the direction that leads

²Following Smith, Mitchell, Chestek, and Buchanan (1977), we distinguish a learning system’s *performance element* (or *performance system*) – the component that actually performs the objective task – from the components that attempt to improve performance.

to the state deemed more desirable. The job of the learning component is to *select* a few relevant states from each run to be memorized and to *label* each as either desirable or undesirable. This credit assignment problem is very complicated, because feedback only comes at the end of a run, and even then it only indicates that the overall run has failed. The mechanism used to decide which states to retain (for both credit and blame) involves several empirically determined parameters.

Many previous researchers have addressed this same inverted-pole problem using techniques ranging from control theory to connectionism. This work improves on those earlier approaches, because it does not depend on a pre-defined partitioning of the continuous state space into discrete regions. Furthermore, in very few trials, it converges to the point that it can keep the pole balanced for thousands of time steps. On several different runs of the overall system, using various different initial settings, CART always converges within 16 trials. (By contrast, the one earlier system that did not need pre-defined discrete regions required thousands of trials.)

Fisher's "Improving Inference through Conceptual Clustering" describes the COBWEB system, which computes conceptual clusters and uses these clusters to predict the values of unseen attributes of new objects. The system is given several completely specified objects in feature vector form. From these training instances, it forms conceptual clusters having a high *category utility* (Gluck & Corter, 1985) - i.e., the categories have many similarities between members of each class, but few similarities between members of different classes. For each of these clusters, COBWEB computes the conditional probabilities for each value of each feature, given membership in that cluster.

The process of making predictions for new objects works as follows. Given a new object for which only *some* of the feature values are specified, COBWEB classifies this new object into one of its clusters. Then, the missing feature values are predicted to be the most commonly occurring values for that cluster. For example, COBWEB can use a set of training examples describing patients and their (correct) diseases to form clusters that, presumably, group patients according to their diseases and symptoms. To diagnose a new patient, x , COBWEB uses x 's symptoms to classify x into the proper cluster and then predicts x 's missing feature - the disease, in this case - to be the most common value of this cluster.

This article first motivates its use of category utilities as a way of forming these useful categories - useful for predicting the unspecified values of new objects - and then outlines the particular heuristics COBWEB uses to find these categories. It also provides empirical evidence that the system works effectively by showing that COBWEB can diagnose Stepp's (1984) full range of soybean diseases perfectly, after using only ten training instances to form the categories! It also illustrates how tightly certain features of this particular data set are correlated with one another - especially diagnosis with symptoms - to explain this extremely fast convergence.

Schlimmer's "Learning and Representation Change" describes how the STAGGER concept acquisition system can augment its initial concept language when necessary. Its performance system (Schlimmer & Granger, 1986) computes the odds that a given object belongs to a particular class based on the object's known features. This article describes how STAGGER's three learning modules incrementally learn this feature-set-to-class odds function from a series of training examples.

- (1) The weight-learning module examines how often each feature is, and is not, associated with each class. It uses these numbers to compute how strongly the presence, or absence, of this feature in some object, x , affects the odds that x belongs to that class. As this module behaves like a perceptron (Minsky & Papert, 1969), there are certain classifications that it, alone, will never be able to learn.
- (2) STAGGER's Boolean learning module attempts to overcome this limitation by forming new features as Boolean combinations of existing ones. It considers forming these new features whenever the first module has misclassified some training example.³
- (3) STAGGER's numerical learning module deals with continuously-valued attributes. Each example supplies an actual value for this attribute; this set of values forms the possible boundary points. (E.g., if the value of some example's A_i attribute is v , then this module considers partitioning the values of A_i into the values greater than v and the values less than v .) It then uses a utility measure to decide which subset of these values form the boundary points of a useful set of "buckets."

The article first describes these components and indicates how they fit together. The latter two modules learn new representations for the examples by forming new "Boolean features" as Boolean combinations of existing ones and new "discretized features" as intervals over continuous ones. The first module can then weight these new features along with the initial ones. The article also provides a simple example of STAGGER's behavior and closes by describing some of the system's inherent limitations.

We conclude this section with a few general comments on these papers. Each paper describes an interesting technique for finding a meaningful classification of a set of examples. These systems all fit within Quinlan's criteria; their differences lie in whether the classes are pre-defined, whether the training examples are explicitly given and explicitly labeled, and whether the representations of examples can change. CART uses a series of experiments to learn how to map a set of pre-defined attributes into the two pre-defined classes: left-push versus right-push. However, it must select the relevant training examples from the set of all observed states, and it must label them appropriately. By contrast, COBWEB is given explicitly defined training examples in a fixed, pre-defined representation. It must form the meaningful clusters and then decide how to assign partially specified examples to the appropriate class. STAGGER is also

³That is, STAGGER incrementally examines a given sequence of training examples and their respective classifications. The weight-learning module proposes a class for each of these examples; this is considered a "misclassification" if it differs from the given "correct" class.

given a pre-defined set of classes and an explicitly labeled series of training examples. However, the features given in the examples are not sufficient to let STAGGER's very simple learning algorithm (weight learning) find correct concept definitions, so it must introduce new features to overcome this problem.

All three of these papers presented some validation of their research. The first two provide strong empirical evidence that their respective approaches work effectively by demonstrating that they are sufficient to solve some recognized challenges. The third paper attempts a theoretical sufficiency proof - describing when its technique is guaranteed to work.

3. Explanation-based learning techniques

Recently, there has been growing interest in machine learning systems that learn how to solve problems more efficiently. Such systems begin with a rich domain theory, usually one sufficient to allow the system to solve the posed problem when given arbitrarily large amounts of computation time and space. The objective of such "explanation-based learning" (EBL) algorithms is to modify the performance system to allow it to find the solution faster.

An EBL system begins by explaining a given solution to a specific query; this often involves examining the proof tree (or explanation structure) that was constructed to solve this query. It then uses a form of goal regression to determine some set of conditions that is sufficient to derive the query. (These are often the leaf nodes of the proof tree.) Finally, it abstracts away the details specific to this particular query, leaving a set of conditions that are provably sufficient to solve a more general class of queries. This information constitutes a new "chunk" or "compiled rule," which is then added to the system's knowledge base. The performance system can then use this chunk to solve future problems, often more efficiently (Mitchell, Keller, & Kedar-Cabelli, 1986; DeJong & Mooney, 1986).

All four papers reviewed in this section deal with EBL systems. The first two papers describe how specific EBL systems can extend and generalize a given explanation, and the latter two describe how to characterize and generalize the output of an EBL system.

Prieditis and Mostow's "Towards a PROLOG Interpreter that Learns" describes PROLEARN, an adaptive PROLOG interpreter. Like most EBL systems, this one first answers a query, and then employs a particular form of EBL to generalize the execution trace. PROLEARN then applies partial evaluation to simplify the resulting generalizations into more efficient subroutines. Finally, these simplified generalized subroutines are added to the PROLOG program for future use.

One example concerns the Towers of Hanoi puzzle. After solving the puzzle with three disks, PROLEARN's learning process produces a new clause that is, by itself, sufficient to solve the three-disk problem. This clause is asserted "before" the general Hanoi clause so that PROLEARN will use it first in the future. This newly learned clause considerably reduces the system's execution time on related problems, such as the eight-disk one.

After presenting this approach, the authors discuss some of its inherent problems. One of the major difficulties is that including a newly learned subroutine can sometimes slow down the system's performance, as it forces PROLEARN to consider special-case clauses that may not apply to the current query. This is a recognized problem: it is not necessarily desirable to retain everything that has been learned (Minton, 1985).

Shavlik and DeJong's "BAGGER: A System that Extends and Generalizes Explanations" describes a learning system that can generalize the structure of an explanation by generalizing the number of times an inference rule can be used. BAGGER analyzes the explanation structure, looking for repeated, interdependent applications of rules. When the system finds such substructures, it forms a new rule by first extending the explanation to allow for an arbitrary number of repeated applications of the rule sequence and then generalizing using a standard EBL process. BAGGER only incorporates *operational* new rules – that is, rules whose preconditions depend only on the original state and not on the results of any intermediate applications of the original rule.

As one demonstration, BAGGER is given an example of unstacking a tower of three blocks, to achieve the goal of exposing the bottom block. It generalizes this into a plan to expose an arbitrary block in a tower of arbitrary height. This generalization is beyond the abilities of a "classical" EBL system.

EBL systems seek a set of conditions that are sufficient to describe some concept. There can, however, be many different descriptions for any given concept. Kedar-Cabelli's "Formulating Concepts According to Purpose" presents an EBL system, PURFORM, that uses an agent's purpose to select an appropriate concept description. For example, PURFORM could describe a cup one way to an agent trying to satisfy its thirst and a different way to an agent seeking an ornament.

PURFORM is given a plan that satisfies a certain goal (e.g., a plan that satisfies the goal of quenching an agent's thirst). It uses EBL-style goal regression to find the general initial preconditions of the plan. The goal regression process is focused by a role-analysis procedure that attempts to understand how the artifact (e.g., the cup) fits into the initial preconditions. It employs two pruning heuristics to delete properties from this conjunction of preconditions, leaving only properties that are *relevant* (i.e., that include the artifact as one of their arguments) and *intrinsic* (i.e., that do not appear on the add or delete list of any operator). Unfortunately, as the paper describes only one example and only one purpose, it is hard to evaluate either the general ideas or these particular heuristics.

In his award-winning paper, "Defining Operationality for Explanation-Based Learning," Keller examines a standard definition of operationality: "a concept description is operational if it can be used efficiently to recognize instances of the concept it denotes." He explains how this is implemented in several EBL systems and points out what is wrong with the definition. He also identifies three dimensions for discussing operationality: variability, granularity, and certainty.

His major point is that the “efficient instance recognition” definition makes too many restrictive assumptions about how the concept description will be used to improve performance. For example, how is efficiency defined? Although CPU time is commonly used, this may not be a good metric; e.g., perhaps space efficiency would be better. Keller’s solution involves giving the learning algorithm knowledge of the performance system and providing explicit performance objectives. His METALEX program embodies the revised definition of operationality. He also notes that METALEX’s particular technique can be extremely slow, which may limit its practical application.

We close this section with a general comment on this sub-subfield. While EBL is a fairly recent development in machine learning, we have recently seen a great many papers on this topic, some proposing variations of the algorithm and others examining the existing algorithms in detail, sometimes uncovering errors in previous work. Machine learning (as well as AI in general) will certainly benefit from more of this kind of focused research, in which many diverse researchers explore important details in a common area.

4. Theoretical results

This section describes three papers that make contributions of a predominantly theoretical nature. The first two offer analyses of concept learning methods, whereas the third describes how the SOAR system can incorporate new information from external sources.

Haussler’s “Learning Conjunctive Concepts in Structural Domains” examines algorithms for learning conjunctive concepts, viewing this as a search through a predefined space of candidate concepts, seeking one that is consistent with the training examples. This research provides a theoretical analysis of complexity in two learning paradigms – Mitchell’s (1979) candidate elimination approach and Valiant’s (1984, 1985) framework. Haussler’s key theorem demonstrates that candidate elimination learning can be exponential in the number of attributes used to describe each object in an example. He also proposes several ways of side-stepping this negative result and then proves that these strategies have severe limitations for non-trivial examples (e.g., those with more than a few objects).

Another result deals with “simple” hypotheses (i.e., those whose formulae require few atoms) that are consistent with the observed training examples. Haussler proves that, with a large enough set of examples, such simple hypotheses tend to be good approximations to the target concept – that is, they are unlikely to misclassify a subsequently drawn random example. This corroborates Occam’s Razor and suggests that we should have more confidence in simple hypotheses.

In “A Declarative Approach to Bias in Concept Learning,” Russell and Grosz view the process of learning a concept from examples as an inference process:

$$\text{bias} + \text{observations} \mapsto \text{hypothesis}$$

where both the “bias” (i.e., the initial collection of known information) and the observations (read “instance descriptions”) are declaratively expressed as propositions in first-order logic. The process of updating the version space can now be described as a deduction from the bias and examples. As the bias limits the set of concepts the candidate elimination algorithm can express,⁴ it is critical to find a good bias. This paper addresses two aspects of this challenge: how to set up the initial bias, and how to shift from one bias to another.

The paper first shows how a learning system can use background knowledge to derive an initial bias and, in particular, to define the languages used to describe the instances and concept candidates. It illustrates how this bias could be used to define the initial version space used by the META-DENDRAL system. The second half of the paper shows how to express shifts in version-space bias as deduction in a non-monotonic formalism of prioritized defaults. It then shows that this formalism is sufficient to represent the standard bias shifts (viz., of weakening a strong bias when the system’s conclusions are inconsistent with observations), as well as others. This approach generalizes the candidate elimination algorithm by allowing it to incorporate arbitrary background knowledge and by showing how it could be applied in less structured domains.

Rosenbloom, Laird, and Newell’s “Knowledge Level Learning in SOAR” describes the generality of the SOAR system. Earlier research (e.g., Laird, Newell, & Rosenbloom, 1986) has demonstrated that this system can create new productions, or “chunks,” based on the results of its goal-based problem solving and then use these chunks to speed up its performance on subsequent goals. Because each new chunk is logically entailed by the pre-existing knowledge base, this technique has been considered symbol-level learning (Dietterich, 1986, Newell, 1981).

This paper describes how SOAR can use this same chunking mechanism to learn at the *knowledge level* - that is, to acquire new knowledge. The authors consider a special case, which they call “data chunking,” that involves observing new information from the outside and then storing some representation of this new knowledge in memory. Learning occurs when a new chunk is created that can retrieve the new knowledge and add it to working memory. The key is for SOAR to set up the appropriate internal tasks, ones that allow its problem-solving process to create chunks that represent the new knowledge. The article demonstrates this knowledge-level chunking in two declarative memory tasks: recognition and recall of new objects.

To summarize these articles: many learning systems must make inductive leaps to form conclusions. Haussler shows that the problem is only effectively solvable (by the candidate-elimination algorithm) on small tasks. As consistency with observed instances is not enough, each system must embody an implicit bias to constrain the possible hypotheses it can produce. This is precisely the kind of information that Russell and Grosz capture in their non-

⁴I.e., a candidate elimination system using the wrong bias will be unable to express the goal concept and so will be unable to return any generalization of the examples.

monotonic formalism for shifting bias. Another fundamental problem is how to represent the information derived from observations as new knowledge in the theory. Rosenbloom, Laird, and Newell address this issue, describing one way of extending a theory through perception.

5. Knowledge acquisition

Knowledge acquisition (KA), like machine learning, describes techniques for increasing the functionality of a computer system. KA focuses on the identification and representation of knowledge for use in expert systems. Since this knowledge can be acquired in many ways – for example, by interviewing human experts, analyzing test cases, and refining existing knowledge – a wide variety of techniques have been studied. The three KA papers at AAAI-87 represent different approaches, namely, comparing two methods of encoding a human expert's knowledge, automatically extracting an expert's knowledge, and automatically constructing optimally predictive rules from test cases.

Henrion and Cooley's "An Experimental Comparison of Knowledge Engineering for Expert Systems and For Decision Analysis" contrasts the knowledge acquisition process for a traditional rule-based expert system with the process of building a probabilistic model using decision analysis techniques. To study the two approaches, they had an experienced knowledge engineer and a decision analyst each interview a plant pathologist and design systems for the same diagnostic task. Each approach involves identifying the domain objects and building inference graphs from them. In the expert system model, arcs in the graph represent evidential links, based on the domain expert's reasoning heuristics. Each inference step in the graph can be directly translated into a rule. Uncertainty is represented as the expert's degree of belief in a rule conclusion (diagnosis) given the premises. In the decision analysis model, a Bayesian belief net (also called an influence diagram) is constructed; here, the arcs represent the believed direction of causality. The expert must estimate the conditional probability of each consequent given its antecedents.

Although the expert system and decision analysis approaches resulted in superficially similar models of the domain, there were several important differences in terms of the model-building processes, as well as in the resulting systems. The expert system approach, which encodes heuristic knowledge in a less formal language, is found to require less effort from a knowledge engineering perspective. However, it is more likely to result in unexpected results, thus requiring more testing and refinement. The decision analysis approach, in contrast, requires a much more rigorous analysis of all possible combinations of conditions and the estimation of their probabilities; a number of new techniques for making this process more tractable are presented in the paper. Some of the advantages of the decision analysis approach are its flexible and powerful inferencing capabilities and, because of its rigorous probabilistic model of the domain, its potential for improving on the relatively limited, intuitive reasoning capabilities of a human expert.

Klinker, Boyd, Genetet, and McDermott's paper, "A KNACK for Knowledge Acquisition," describes an automatic knowledge acquisition system called

KNACK. Through interactions with an expert in the domain of electromechanical systems, who is not familiar with expert system construction, this system builds an expert system that is capable of evaluating specific designs.

KNACK proceeds through a number of stages of knowledge acquisition, steps that are similar to the conventional approach that a (human) knowledge engineer would take. The system analyzes a domain expert's design report and, using its own knowledge about evaluating designs, builds a conceptual model of the domain, including relevant concepts and vocabulary. It next analyzes the expert's report in greater detail and, through further interaction with the expert, forms more generally applicable abstracts from fragments of the report. Specific instances of these fragments are displayed and can be edited by the expert. Finally, it uses heuristics to detect gaps in the knowledge base.

Knowledge acquisition for expert systems is typically a process of recording the informal, intuitive heuristics used by an expert, and then iteratively refining this knowledge through trial and error. In "Optimizing the Predictive Value of Diagnostic Decision Rules," Weiss, Galen, and Tadepalli examine one way of formalizing this process. They describe an approach to generating optimally predictive diagnostic rules based on case analyses. Diagnosis is treated as a standard statistical decision-making problem; thus rule formation involves finding a combination of tests (which will become a rule premise) that have a high positive predictive value (i.e., where a positive test outcome is highly correlated with a correct diagnosis).

The article presents a heuristic search algorithm that finds a near-optimal combination of tests (in terms of predictive power) for some diagnosis. The algorithm first examines how the values for each individual feature are correlated with the diagnostic hypothesis and uses this to suggest "interesting" cut-off points in the range of possible values. It then considers Boolean combinations of (up to a fixed number of) these `feature < cutoff` tests, seeking combinations that are optimal in terms of positive predictive power, while also satisfying other statistical constraints.

The article discusses various heuristics used to prune the space of possible rules. It empirically demonstrates their effectiveness by showing that this algorithm is able to construct rules as good as those produced by human experts by analyzing a medical data base of about 3,000 cases. It also discusses further applications of this technique, including applying it to refine an existing rule base and to statistically predict future performance in situations where only an impoverished or unrepresentative sample of test cases is available.

To conclude this section on knowledge acquisition, we note that although practical expert systems continue to be developed using standard knowledge engineering techniques, the trend in research in this field has been toward increasingly automated systems, using increasingly formal techniques. Systems such as KNACK have been successful at automating parts of this knowledge acquisition process. More formal and rigorous approaches, such as the decision analysis model described by Henrion and Cooley and the automatic construction of rules from test cases as described by Weiss et al., show promise for future research in knowledge acquisition.

6. Other topics

There were many papers that had a definite “machine learning” theme (based on our view of this sub-area), but that were classified into other sub-fields. This section provides pointers to some of the relevant papers.

6.1 Analogy

Analogical reasoning is traditionally a sub-area of ML. Gentner, a psychologist, presented a very thorough survey of this topic – covering more than her title, “Cognitive Modeling of Analogy,” would suggest. After presenting her “dimensions” for the analogy process, she succinctly discussed the myriad of recent work in this area, in each case fitting that research within her model. She also described her experiments with human subjects, to provide psychological evidence for various claims. For example, she summarized evidence that the mechanism for *accessing* an analogy (i.e., going from a “target problem” to the relevant target and base analogues) is different from the mechanism that uses the analogy (i.e., that *uses* the target and base analogues to understand more about the target analogue or to solve the target problem).

Various other papers dealt with analogy. For example, one cognitive modeling session was devoted to “Access and Analogy.” (It included Skorstad, Falkenhainer, and Gentner’s paper, which describes an implementation of Gentner’s ideas.) The automated reasoning paper by de la Tour and Caferra discusses how to use analogies when proving theorems.

6.2 Improving efficiency

Machine learning now encompasses algorithms that attempt to reformulate a problem, often to speed up a computation. Many other sub-fields of AI have addressed this problem. As this also characterizes the formation of macro-operators, many results from the search and planning areas are relevant. (Korf mentions this in his excellent “Search” survey.) Similarly, automatic programming involves changing an inefficient specification of a process into an effective one. Papers on this topic included the automated reasoning papers by Kelly, by McCartney, and by Manna and Waldinger, and the knowledge representation paper by Lowry. (Many of these systems must also extend the information given by making educated guesses – see Subsection 6.3 below.) The major obstacle addressed by the automated deduction and theorem proving community is finding ways of speeding up a proof. (See both Bledsoe’s “Automated Deduction” survey and the above-mentioned de la Tour and Caferra paper.)

6.3 Theory extension and revision

Given a corpus of information, one can deduce a well-defined set of conclusions. This is seldom sufficient – one often needs to *extend* the given “theory” by making educated guesses (e.g., default assumptions), or to *revise* this theory, by removing some inconsistencies (Greiner, 1987). Issues related to these processes have recently become a major focus of current research in knowledge representation and reasoning: there were over 20 other papers (in addi-

tion to the ML and automatic programming papers mentioned above) in this conference related to this theme, including sixteen papers dealing explicitly with issues in default reasoning and six others on technical issues in temporal reasoning (including Shoham's "Temporal Reasoning" survey). These papers ranged from theoretical descriptions of this abstract process, through empirically collected observations based on specific implementations, and included three prize-winning papers by Dean and Boddy, Delgrande, and Morris.

6.4 Cognitive modeling

Obviously, "machine learning" has close ties with "human learning," whose intersection with AI falls within the category of cognitive modeling. There were several such papers presented, including the (above-mentioned) "Access and Analogy" session, Ballard's paper on connectionistic learning, and Drescher's paper on early Piagetian learning.

7. Conclusion

This concluding section discusses first the trends suggested by these papers and then some of the general issues they raise.

7.1 Trends in machine learning

Nearly all subareas of machine learning were well represented in this conference. These papers suggest two apparent trends in ML, evident across these sub-topics. First, most of the papers dealt explicitly with symbolic knowledge-based reasoning, in the form of an explicit domain theory (in all EBL systems), explicit biases (Russell and Grosz), and so on.

The second trend is even more pervasive: the ML community has become increasingly concerned about validating claims and demonstrating solid research results. This has led to an increased emphasis on both theoretical statements and on careful empirical validations. Particularly significant are the experimental comparisons of different systems working with the same training data. For example, Connell and Utgoff, Fisher, and Weiss et al. each compared their (respective) system's performance with others that have attempted the same task.

7.2 Issues for machine learning

This conference also raised many issues, including the imprecise definition of machine learning and its role in these general AI conferences. While ML has recently begun to develop its own identity, it still has a sizable overlap with a variety of other AI concerns, as discussed in Section 6. Artificial intelligence, as a whole, would certainly benefit from a more precise understanding of these inter-connections, as it would help researchers to understand how their work relates to others in seemingly disparate areas.

This understanding might also help to address another important concern. ML has matured to the point that it now has its own conference (the International Conference on Machine Learning). As papers may not appear in both

an ML conference and a general AI conference (i.e., an AAAI or an IJCAI), researchers will have to decide which audience they want their results to reach. Knowing which “ML issues” are important to other AI researchers will help future researchers decide which audience will benefit most from their results and, hence, which conference is most appropriate.

Finally, we close this review with a recommendation that ties together several of the themes mentioned in this section. The call for papers for the Machine Learning Conference required every submission to include an explicit “evaluation criterion” on which it should be judged. We recommend that the general AI conferences follow suit.

Acknowledgements

The authors thank Tom Dietterich, Ken Forbus, Evangelos Milios, Calvin Ostrum, and Bart Selman for their helpful comments. R. Greiner and S. Becker were supported by a grant from Canada’s Natural Science and Engineering Research Council, and B. Silver was supported by his employer, GTE Laboratories Incorporated.

References

- Dietterich, T. G. (1986). Learning at the knowledge level. *Machine Learning*, 1, 287-315.
- DeJong, G., & Mooney, R. (1986). Explanation-based learning: An alternative view. *Machine Learning*, 1, 145-176.
- Gluck, M., & Corter, J. (1985). Information, uncertainty, and the utility of categories. *Proceedings of the Seventh Annual Conference of the Cognitive Science Society* (pp. 283-287). Irvine, CA: Lawrence Erlbaum.
- Greiner, R. (1987). *Principles of inference processes* (Technical Report CSRI 193). Toronto, Ontario: University of Toronto, Department of Computer Science.
- Laird, J. E., Newell, A., & Rosenbloom, P. S. (1986). Chunking in SOAR: The anatomy of a general learning mechanism. *Machine Learning*, 1, 11-46.
- Minsky, M. & Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.
- Minton, S. (1985). Selectively generalizing plans for problem solving. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 596-599). Los Angeles, CA: Morgan Kaufmann.
- Mitchell, T. M. (1977). Version spaces: A candidate elimination approach to rule learning. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (pp. 305-310). Massachusetts Institute of Technology: Morgan Kaufmann.
- Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. T. (1986). Example-based generalization: A unifying view. *Machine Learning*, 1, 47-80.
- Newell, A. (1981). The knowledge level. *AI Magazine*, 2, 1-20.
- Schlimmer, J. C., & Granger, R. H. (1986). Incremental learning from noisy data. *Machine Learning*, 1, 317-354.

- Smith, R. G., Mitchell, T. M., Chestek, R., & Buchanan, B. G. (1977). A model for learning systems. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (pp. 338–343). Cambridge, MA: Morgan Kaufmann.
- Stepp, R. (1984). *Conjunctive conceptual clustering: A methodology and experimentation*. Doctoral dissertation, Department of Computer Science, University of Illinois, Urbana.
- Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27, 1134–1142.
- Valiant, L. G. (1985). Learning disjunctions of conjunctions. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 560–566). Los Angeles, CA: Morgan Kaufmann.