SEVEN DESIRABLE PROPERTIES FOR ARTIFICIAL LEARNING SYSTEMS

Christophe Giraud-Carrier and Tony Martinez
Brigham Young University, Department of Computer Science, Provo, UT 84602
cgc@axon.cs.byu.edu, martinez@cs.byu.edu

Abstract

Much effort has been devoted to understanding learning and reasoning in artificial intelligence, giving rise to a wide collection of models. For the most part, these models focus on some observed characteristic of human learning, such as induction or analogy, in an effort to emulate (and possibly exceed) human abilities. We propose seven desirable properties for artificial learning systems: incrementality, non-monotonicity, inconsistency and conflicting defaults handling, abstraction, self-organization, generalization, and computational tractability. We examine each of these properties in turn and show how their (combined) use can improve learning and reasoning, as well as potentially widen the range of applications of artificial learning systems. An overview of the algorithm PDL2, that begins to integrate the above properties, is given as a proof of concept.

1. INTRODUCTION

Over the last 30 years, extensive work has been done in the area of Machine Learning. Whether the approach is inspired by biological plausibility (Artificial Neural Networks) or by psychological plausibility (Artificial Intelligence), the goal is to design systems capable of learning and reasoning about certain tasks (e.g., analogy, classification, etc.). Most existing systems focus on some observed aspect of human learning, and attempt to match, and possibly exceed, human abilities. For example, several inductive models exist that exhibit similar (and sometimes better) predictive accuracy than their human counterparts on various problems (e.g., loan underwriting, mine-rock discrimination). It is a system's ability to capture and exhibit certain important characteristics (often inspired by human learning), that allows it to be useful as an artificial learning system.

We propose seven desirable properties for artificial learning systems, namely: incrementality, non-monotonicity, inconsistency and conflicting defaults handling, abstraction, self-organization, generalization, and computational tractability. This proposed set of properties is not claimed to be complete, nor does it imply that all properties must necessarily be present in all learning models. Rather, it focuses on issues that are not often explicitly addressed and provides a basic foundation for the design of more efficient algorithms. As most other computer-based systems, learning systems are difficult to reverse engineer, and desirable properties should be part of the original design rather than retrofitted into the system after the fact.

In the following sections, we examine each property in turn and show how their use can improve learning and reasoning, and potentially widen the range of applications of artificial learning systems. As a proof of concept, the algorithm PDL2 [5] is overviewed. PDL2 begins to integrate the above properties into a unified framework in which inductive learning supplements the use of prior knowledge to yield a model capable of dealing with a greater variety of interesting problems.

2. INCREMENTALITY

Humans acquire knowledge *incrementally*, that is, they learn over time. All the information necessary to learn a given concept is rarely available a priori. Rather, new pieces of information become available over time, and the knowledge base is constantly revised (i.e., evolves) based on the newly acquired information. To have all of the knowledge a priori amounts to programming rather than learning. One would then have to assume the existence of a "smart" teacher that can program the system to perform the proposed task. Unless the teacher is another machine, such an assumption seems to go against the long term goal of building artificial systems capable of learning. Moreover, in the context of learning grammar with a recurrent network, Elman [2]

also gives empirical evidence that the "network fails to learn the task when the entire data set is presented all at once, but succeeds when the data are presented incrementally."

Another aspect of incrementality has to do with the form in which knowledge is acquired. Because of built-in mechanisms (e.g., pain) and social structures (e.g., the family, school), humans are able to combine both specific examples and more general rules to efficiently learn complex problems. For example, as shown in [4], it is common that humans first get exposed to general rules and then to exceptions. Elman [2] suggests that the same form of incrementality may actually be the result of an early handicap (e.g., limited memory and attention span characteristic of children). In other words, it may be that the learning system itself evolves over time, rather than the knowledge it is presented.

In either case, the ability to use rules and examples increases learning speed by pruning and constraining the search in the input space, reduces memory requirements, and improves overall predictive accuracy (see, for example [4]). Moreover, with appropriate generalization mechanisms, examples can effectively supplement rules given a priori by generating new rules, and modifying existing ones. Hence, the system combines the intensional approach (based on features, expressed by rules) and the extensional approach (based on instances, expressed by examples). Such a combination may prove useful in commonsense reasoning. In particular, it allows the system to retain the self-adaptivity inherent in human learning, while not having to unnecessarily suffer from poor or atypical learning environments.

Chronology is inherent in incrementality. Thus it is also possible to implicitly integrate time as a factor and another source of bias in learning. For example, the system could choose to give precedence to newly acquired knowledge, recognizing the possibility that early guesses may be incorrect (see, for example [7]). In an iterated training set system, the distribution of "weight" over the various pieces of knowledge is uniform. So, a temporal ordering of the various pieces of knowledge would have to be made explicit (most likely by some external teacher) to achieve the same result.

Note also that if only minor revisions are made to the current knowledge base as new information is gained, and no major reprocessing of previous observations need be done, then an incremental system may allow a reduction of overall algorithmic complexity.

3. NON-MONOTONICITY

Classical first-order logic is monotonic, that is, if *A* is a set of axioms and *C* is derivable from *A*, then *C* is also derivable from any superset of *A*. Informally, this says that the acquisition of new knowledge can never invalidate previously known facts. Though theoretically convenient, monotonicity certainly does not hold in everyday life, where information is often uncertain and incomplete. Consider the following classical example. Suppose I tell you that Tweety is a bird and ask you whether Tweety can fly. Since you know that birds typically fly, and you have no reason to believe otherwise concerning Tweety, you are most likely to tell me that Tweety can fly. Now, imagine that later on I tell you that Tweety is a penguin. Then, since you know that penguins, though they are birds, do not fly, you will most likely withdraw your previous conclusion and tell me that Tweety cannot fly. The new piece of knowledge you gained invalidated the truth of the previously accepted fact.

Such patterns of reasoning, often equated with commonsense reasoning by logicians (see, for example [1, 6]), are pervasive in the way humans deal with the inherent uncertainty and incompleteness of the world (or their representation thereof). The development of systems that effectively handle classical commonsense protocols, such as inheritance, is a first step in overcoming the brittleness bottleneck.

Non-monotonicity results from incrementality. If all the knowledge is available a priori, then non-monotonicity is not an issue. That is, there is no need for special learning mechanisms that invalidate portions of knowledge, while not affecting the rest of it. Essentially, the world is closed. If knowledge is incrementally made available, however, then more work is required. Consider the above example again. Knowing that Tweety is a penguin should only directly and explicitly affect knowledge about Tweety, not any other birds. Just because I now know that Tweety is a penguin and that penguins do not fly, I (most likely) would not want to conclude that Fred (another bird that I encounter later) does not fly, even though my belief in Fred's flying ability may be (implicitly) lessened by my having discovered the existence of exceptions to the rule.

4. INCONSISTENCY AND CONFLICTING DEFAULTS HANDLING

Everyday life is replete with inconsistencies and conflicting defaults, such as the famous Nixon Diamond [9]. Given that Quakers are typically pacifist, that Republicans are typically not pacifist, and that Nixon is both a Republican and a Quaker, what can you say about Nixon's dispositions? Removing inconsistency does simplify the learning task, but is usually artificial and does not reflect the world. Indeed, as pointed out by Szepesvári and Lörincz [10], internal representations

of the world are bound to be partial so that what we are left to deal with are relations rather than functions. Inconsistency is a fact of life.

In the face of inconsistencies, as in the above example, it is not reasonable (as first-order logic would require) to force the knowledge base to become consistent by arbitrarily ridding it of one (or part) of the offending rules. Instead, while both rules are essentially good as default rules, Nixon appears to be an exception to at least one of them. Mechanisms should be developed to handle such situations. If the extensional and intensional approaches are combined, greater flexibility can be achieved. There are at least three available options: 1) default rules are given a priori, along with their respective priority (e.g., religion is stronger than politics), 2) default rules are given a priori, but their relative priorities are computed dynamically via exposition to a set of examples, or 3) default rules and their priorities are inductively learned by the system. The first option guarantees the expected result (according to some teacher), the second option adds flexibility by letting experience supplement a priori knowledge, and the third option produces a model consistent with current experience with the world. These options represent a trade-off between self-adaptivity of the system and consistency of the system with some external entity's representation of the world.

5. ABSTRACTION

Humans are good at dealing with symbols. Many machine learning techniques are limited by their *attribute-value pair* representation language. Attribute-value pairs make it difficult to deal with open domains and higher-level, abstract relationships. For instance, if the relation to be found is equality of two variables (e.g., learning to discriminate between squares and rectangles), then the algorithm may discover a rule for each value appearing in the training set, but may not be able to extend them to all possible values. This suggests that a more symbolic representation be used so that such high-level relationships can be generated when they exist. The need for such a language is likely to be a consequence of the fact that current artificial systems lack the richness of the human sensory system. The more ways there are to observe and interact with an object (visual, tactile, etc.) the richer its final internal representation.

Symbols also tend to be much easier for human interaction and understanding (see, for example, Michalski's comprehensibility bias). There is however a trade-off between the efficiency of numerical representations and the flexibility of more symbolic representations. Neural nets, which use numbers as inputs, are computationally efficient, but they lack a "comprehensible" interface with the human user. This opacity constitutes one of the main criticisms of most numerical approaches [8]. On the other hand, symbol-manipulating systems are easier to interact with and understand, but their efficient implementation remains a computational challenge. Research is currently under way to design representation languages that take advantage of both approaches.

6. SELF-ORGANIZATION

Because the human brain consists of complex networks of neurons, some advocate artificial neural networks as the best approach to machine learning. Though this position is a little extreme (and too committal), neural networks have many interesting features, in particular self-organization and massive parallelism.

Architectures that are massively parallel offer a platform for the design of more efficient algorithms. If knowledge is distributed and simple nodes are used, even mechanisms as simple as *gather and broadcast* can be used to update the entire current knowledge base at once. The use of parallelism can also often aid the problem of computational complexity.

By self-organization we mean the ability to adapt to new information and to consequently dynamically maintain and evolve the knowledge base. The system must be able to learn the given task, rather than the task be explicitly programmed into the system. Self-organization is inherent in human learning since incrementality forces constant adaptation. From a practical standpoint, this means that if artificial systems are self-organizing, then they can potentially be used in environments that present a hazard to humans (e.g., radioactive sites), or in highly interactive settings in which the machine adapts to its user. Appropriate artificial sensory devices must be developed for such applications.

We would also suggest, from a more psychological standpoint, that the ability to self-organize constitutes a first step in the direction of self-awareness and metarepresentational ability, which appear to be prerequisites to common sense (see [3]). Indeed, self-organization presupposes that the system knows that it knows something. This awareness may in turn be exploited to represent knowledge about knowledge. In fact, if machines can learn, they could potentially learn how to learn, and consequently teach more effectively.

7. GENERALIZATION

The issue of generalization is two-fold. The first aspect is that of the generalization language and what it can

generate, and the second aspect is that of predictive accuracy.

The generalization language, which one can equate with the output space of the learning algorithm, determines the kind of concepts that the system can generate. There is often a mismatch between the representation language and the generalization language. Going back to the equality example of Section 5, one can easily see that adding new special symbols to the attribute-value language would allow it to represent equality of two variables. However, the algorithm may still not be able to efficiently generate that relationship (for example, due to the lack of an appropriate generalization rule). Hence, not only the representation language but also the generalization language must be revised. The value of incrementality and self-organization depends in large measure on the ability to generalize.

The other aspect of generalization is predictive accuracy, that is, the ability, based on what one knows or has been trained on, to make correct predictions about new situations. In an iterated training set approach for example, this implies that convergence on the training set is not as important an issue as is predictive accuracy on the test set. Any viable mechanism should exhibit good generalization performance. Incrementality may improve such performance (see, for example [4]). Predictive accuracy is intimately related to the generalization language, since that language also determines the generality of the generated concepts.

8. COMPUTATIONAL TRACTABILITY

Artificial learning systems can often be computationally expensive. Many of the current learning models have cast the generalization problem as a search problem (through the space of all possible generalizations) and thus are faced with combinatorial explosion (often aided by heuristics). For artificial models to be of any practical use, they must be computationally tractable. We suggest that the algorithms should have at most polynomial-time complexity. Parallel, self-organizing, incremental implementations can often reduce complexity.

9. PDL2 - A PROOF OF CONCEPT

The algorithm PDL2 (Precept-Driven Learning), introduced in [5] is an attempt at integrating many of the above properties into a unified framework. PDL2 is an incremental, inductive learning system. It effectively combines examples drawn from experience with the world (i.e., a training set), and a priori knowledge (called *precepts*) obtained from some external source. The combination allows the integration of both intensional and extensional approaches to reasoning.

PDL2's representation language is the classical attribute-value language, with the addition of a special *don't-care* symbol. Examples have all of their attributes asserted to some value, while precepts contain attributes whose values are don't-care. Precepts can be viewed as rules that result from some instantiation of domain knowledge or commonsense. They can be given a priori, or at any time during learning. Precepts serve as learning biases, and are not necessarily correct. Mechanisms exist to identify and deal with exceptions. Examples from the training set are used to generate new rules, as well as to confirm or refute existing precepts.

We briefly discuss how PDL2 integrates some of the properties discussed here. Learning is incremental. Both examples and precepts are used to train the system. Though the order may affect the outcome, examples and precepts are only seen once (i.e., one-shot learning)

PDL2 handles non-monotonicity naturally. Its execution mode implements default reasoning, where the defaults are the precepts. Exceptions (to precepts or rules) are always kept and given priority in reasoning, thus allowing PDL2 to handle inheritance.

Inconsistency is dealt with by counting examples (or precepts) whose attributes all have the same values, but whose implied outputs are different. Only the one with highest count is retained. Conflicting defaults are handled by either static priorities, given a priori with the defaults, or by dynamic priorities obtained by counting examples that are exceptions to conflicting defaults.

PDL2 implements a network of simple nodes that adapts to newly acquired knowledge by dynamically adding and/or removing nodes in the network. No parameters need to be set a priori, and the final network is the result of a sequence of self-organizing transformations.

The generalization language is identical to the representation language (i.e., attribute-value with don't-cares). Despite its inherent limitations, good predictive accuracy is achieved with the current generalization, that consists of causing an attribute's value to become don't-care (i.e., dropping the condition).

PDL2's learning algorithm requires only a simple gather and broadcast mechanism. Both operations are essentially $O(\log n)$, where n is the number of nodes in the network. Parallelism at the node level is inherent and results in low-order polynomial complexity.

PDL2 was tested on several datasets, both to exercise its learning potential (i.e., its ability to inductively synthesize rules from examples to produce high

predictive accuracy on some test set), and to test its ability to deal with important commonsense protocols, such as inheritance, conflicting defaults, missing information, and the lack of a matching rule. Simulation results demonstrate promise. Research is ongoing to extend the basic model.

10. CONCLUSION

The list of properties discussed here is not meant to be exhaustive. We feel that it is very dynamic in nature, and that, as experience is gained, new properties may emerge. We note also that determining membership in such a set of desirable properties is not a trivial task. Consequently, it may be difficult to evaluate whether the set is complete or even useful. Completeness cannot be guaranteed, however, usefulness may be measured in part by empirical studies. For example, does the new property enhance some other aspect of the learning system (e.g., generalization, range of applications, etc.)? Also, psychological and biological evidences in humans may serve as a guide.

Much work still remains to be done to develop learning systems that effectively use the above properties. The algorithm PDL2, overviewed in Section 9, is an attempt at integrating many of them. Work is ongoing to extend the basic model and develop new algorithms. We are attempting to show that the integration of the properties discussed here can indeed result in significant improvements in learning mechanisms and reasoning performance, and in a widening of the range of applications of artificial learning systems.

Acknowledgment

This research was supported in part by grants from Novell Inc. and WordPerfect Corp.

References

[1] Brewka, G. *Nonmonotonic Reasoning: Logical Foundations of Commonsense*. Cambridge University Press, 1991.

[2] Elman, J.L. Incremental learning, or The importance of starting small. CRL Technical Report 9101, La Jolla, CA: University of California, San Diego, Center for Research in Language, March 1991.

[3] Forguson, L. *Common Sense*. Routledge, 1989.

[4] Giraud-Carrier, C., and Martinez, T.R. Using Precepts to Augment Training Set Learning. In *Proceedings of the 1993 International Conference on Artificial Neural Networks and Expert Systems (ANNES'93)*, 1993, 46-51.

[5] Giraud-Carrier, C., and Martinez, T.R. An Incremental Learning Model for Commonsense Reasoning. Submitted.

[6] Lukaszewicz, W. *Non-Monotonic Reasoning: Formalization of Commonsense Reasoning*. Ellis Horwood Limited, 1990.

[7] Martinez, T.R., Hughes, B., and Campbell, D.M. Priority ASOCS. To appear in *Journal of Artificial Neural Networks*, 1994.

[8] Minsky, M. Logical Vs. Analogical or Symbolic Vs. Connectionist or Neat Vs. Scruffy. *AI Magazine*, 12, 2 (Summer 1991), 34-51.

[9] Reiter, R., and Griscuolo, G. On Interacting Defaults. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981, 270-276.

[10] Szepesvári, C., and Lörincz, A. Behavior of Adaptive Self-Organizing Autonomous Agent Working with Cues and Competing Concepts, June 1993. Submitted.