Primitive Concept Formation

Kevin B. Korb and Colin Thompson
Dept. of Computer Science
Monash University
Clayton, Victoria 3168
Australia
{korb,colint}@bruce.cs.monash.edu.au

"I *have* tasted eggs, certainly," said Alice, who was a very truthful child; "but little girls eat eggs quite as much as serpents do, you know."

"I don't believe it," said the Pigeon; "but if they do, why, then they're a kind of serpent: that's all I can say."

—Lewis Carroll

Abstract. Our goal is to demonstrate the feasibility of an autonomous learning agent by developing means to learn and employ concepts in a primitive machine intelligence which must operate in a real-time, uncertain (noisy) environment. This paper reports on the first steps towards such an agent: the development of an agent, Alice, who starts out with only a primitive set of concepts—corresponding to perceptible attributes of objects in the environment and to her own utility function—and who generates a conceptual structure using cognitively plausible rules of concept formation and refinement, abstracting from the immediate attributes of the mushrooms she finds and their longer-term impact on her utility, in a goal-driven manner. The concept formation rules we have developed are more conservative than such standard methods of concept formation as version space methods and ID3. We suggest that this caution offers a competitive advantage in difficult environments.

1 Introduction

Much effort in artificial intelligence has gone into finding ways of producing smart assistants for humans, automated support for human problem solving; indeed, the expert systems which implement such assistance occupy a central place in the academic and industrial environments in which artificial intelligence flourishes. However, recent interest in *autonomous* intelligence—an intelligence which, once unleashed, does not depend upon continuous human aid and comfort—has grown significantly. On the one hand, intelligence of the dependent variety has revealed some inherent weaknesses—especially its brittleness in the face of novelty. On the other hand, Rodney Brooks' "Creatures" have demonstrated the possibility of a more robust, autonomous intelligence—if also a more representationally constrained intelligence (Brooks, 'Intelligence without Representation,' 1991). We agree with Brooks that in order to demonstrate autonomy we should attack intelligence from the ground up, by developing means of responding to the world starting from a primitive state. We disagree, however, that representations are themselves unnecessary, and we assert that a less primitive intelligence will be achievable only when an agent's representations can adapt to a dynamic environment.

In this paper we introduce Alice, who, using cognitively plausible rules of abstraction and specialization, can conceptualize her environment, learning which objects go together in a class and which of those classes are dangerous and which others beneficial. Her rules of concept formation allow her to build, in a goal-sensitive way, a synchronic (semantic associative) network for identifying and associating classes of objects. She is also able to generate diachronic (causal) links relating these concepts with others and with outcomes (utilities) in Bayesian networks, allowing her to anticipate the effects of her actions and so make intelligent plans.

2 Desiderata for Autonomous Agency

The most basic requirement for an autonomous agent is that it should be able to learn about its world. Without inductive learning there is little chance that an agent will be able to respond appropriately to a complex, dynamic and noisy environment. As autonomous agents must fend for themselves, there should be no assumption that the world be divided neatly into a set of "training" cases and a set of "test" cases. Rather, one must be prepared to respond to unexpected environmental demands at any time; therefore, incremental learning should be preferred, allowing one's accumulated learning, at whatever state of readiness, to be deployed at once. Also, one ought to take advantage of whatever information comes one's way, at any time—i.e., the autonomous agent should be devoted to lifetime learning. In order to ensure that what is being learned is maximally useful, the learning processes should themselves be sensitive to the complex of goals currently active—or, in simpler implementations, to the agent's current utility function—i.e., goal-directed learning (Ram and Hunter, 1992). This last point implies that in the context of concept formation the learning will lie conceptually *between* supervised and unsupervised learning: that is, there will be no external teacher punishing the agent for misclassification, however the utility function (or surrogate) responds to the agent's actions and the environment so that concept formation is based upon goal-relevant outcomes, rather than merely upon the intrinsic similarities of objects.

An autonomous agent should satisfy the following preconditions:

- *Primitive Induction*: The agent should be capable of inductive learning with little or no background knowledge. This is a vital feature for avoiding brittleness when confronted with unanticipated environments (see Korb, forthcoming).
- *Generalization and Specialization*: The agent should have means for building a conceptual hierarchy by generalizing (abstracting) from examples and by specializing (splitting concepts) when necessary.
- *Pragmatics*: The agent should be responsive to changing conditions in the environment, especially changes in the availability of time and other resources.

0-7803-2404-8/94/ $4.00 ©1994 IEEE

- *Incremental Learning*: The agent should be able to take advantage of information when it arrives in order to be able to respond to any immediate demands of the environment.

- *Goal-driven Learning*: Learning strategies should be responsive to the agent's changing goals and through them to the agent's changing environment.

- *Defeasibility*: The agent should be able to revise its beliefs when conditions have changed or when it has made some inductive error.

- *Uncertainty*: The agent should be able to learn from noisy data when the noise is due either to measurement error or to a stochastic environment.

3 Alice's Architecture

Alice begins her life with no information about Wonderland and only a few rules with which to cope—but also with an inherent curiosity about the mushrooms and such around her and an inability to avoid wandering around and sampling them. Alice has a number of detectors (primarily visual) that allow her to discover a variety of mushrooms (e.g., color and shape detectors) and a size detector that can aid her in distinguishing mushrooms from, say, duchesses and Cheshire cats. She also has a utility detector that allows her to know the effect of certain actions (such as eating blue mushrooms or getting too close to duchesses). Alice's curiosity is implemented as a goal to discover the utility of various acts and to move around and investigate her environment. The curiosity goal, while never entirely disappearing, is subsequently largely displaced by the more mature goal of maximizing her utility, somewhat as a child's early general learning often gives way to more income-oriented learning later in life.

While satisfying these goals, Alice builds a representation of her environment. Following Holland, et al. (1986) ***Induction***, this learning has two aspects. The first is the formation of synchronic links from detectors to concepts; e.g., Alice may decide that blue, small and spotted objects are a *kind*. The second aspect is the development of diachronic links which describe what happens given the instantiation of some combination of concepts (including effector concepts). For example, Alice may discover that if she eats a blue spotted mushroom (instantiating the nodes of blue, spotted mushrooms and of eating) then she will feel sick. Indeed, it is the very discovery of such an effect that leads to the formation of the concept—it is the relevance of some constellation of features to a goal (maximizing utility) that triggers the goal-driven learning of conceptual structure (see M. desJardins, 1992).

As Holland, et al. make clear, a conceptual hierarchy (or, a default hierarchy) must both aggregate environmental states into categories when they are appropriately similar and refine such categories into subcategories in response to failed expectations. That is, we may say, any inductive system which can dynamically build a conceptual structure must satisfy the principles of generalization and specialization. Alice's concept formation rules support both of these. Suppose Alice eats a mushroom which subsequently increases her utility. She will want to form a concept which records this fact, so that she may repeat the experience. When she ate the mushroom all of her detectors will have been in some state or other; some will have recorded relevant features of the mushroom (say, that it is blue and plain), others irrelevant features of the mushroom (that it is small) and others irrelevant features of the environment at large (for example, nearby sounds). Alice's task, of course, is to identify the relevant kind of mushroom using the features that are predictive. She does this by taking a subset of features as salient and attempting to identify other mushrooms that share those features and, importantly, the utility outcome. Such methods are detailed below.

The second variety of concept formation rule is specialization. The main rule we are using of this type is the exception rule (much discussed by Holland et al., 1986): if a prediction fails—say a mushroom in a category thought to be nourishing turns out to be harmful—look around for a feature of this mushroom that distinguishes it from others in the category. Assuming such a feature is found, a subcategory is created with that newly noticed feature identified as a salient distinguishing attribute of the subcategory and with a new diachronic link identifying its unpleasant after-effects.

These are the basic tools which we have thus far provided Alice. She also has a simple set of effectors for a simple environment: she has the ability to move to one of eight adjacent cells and an eating effector that allows her to munch on any object in her current cell.

Alice's abilities, although limited, allow for a natural extension to abilities which may satisfy the desiderata of section 2. In particular, whereas Alice currently generalizes and specializes concepts based upon a deterministic reading of her environment, we anticipate introducing statistical criteria for her concept formation rules in the future. For example, her environment may include only a few blue mushrooms that are harmful (decreasing utility) among the many that are beneficial (increasing utility), and her perceptual equipment may be insufficient for her to be able to detect any distinguishing feature. At present such a situation would lead to a collapse of her ability to develop conceptual structure, but statistically based rules of concept formation are planned.

4 Methods of Concept Formation

Our intention in developing Alice's architecture, as described above, was to implement the general principles of induction described in Holland, et al. (1986). Their goal was to characterize the necessary features which any intelligent agent capable of learning about its environment must have. As a preliminary to any more sophisticated learning, the agent needs to be able to form concepts that are indicative of the environment in which it is placed. Hence the principles of generalization and specialization. Any adequate methods of concept formation must abide by these principles—that is, they must provide means of generalizing and specializing concepts, whether explicitly or implicitly. Indeed, the classical models of concept formation do provide such means, and it is most revealing to compare how Alice classifies mushrooms with how version space methods and ID3 would proceed.

Version-space methods (Mitchell, 1977) look at positive and negative instances and attempt to define convex (rectangular) areas of the instance space that capture positive instances and avoid negative instances. They proceed explicitly by generalizing to cover positive instances and specializing to remove negative instances as they are presented (i.e., incrementally). Thus given the following instances

```
color....red.....green...blue....red.....red
spots....0.......1.......2.......2.......3
utility..+1......-1......-1......+1......+1
```

a version-space algorithm readily discovers that red mushrooms are nourishing (have positive utility). The underlying assumption that the concepts are convex is useful, being commonly true; furthermore, *some* simplifying assumption is certainly necessary, since the space of possible concepts is exponential. This and related methods, however, are insufficiently flexible: it is difficult to modify the convexity assumption when evidence grows that it is not working; and they are intolerant of noisy data.

ID3 (or C4.5) produces very compact decision (classification) trees using an information-theoretic criterion for selecting splitting attributes (see Quinlan, 1983). This and other means of generating decision trees and graphs (e.g., Oliver, Dowe, and Wallace, 1992) often produce near optimal means for categorizing objects. Furthermore, they do work effectively with noisy data. From the point of view of Alice's prospective environment, though, these techniques have a number of drawbacks. Most notably, they require that a very large sample of objects be available at the beginning as training data. Although they can be modified to deal with new data, they are not designed for incremental learning problems.

In section 6 below we will compare these methods with Alice on some sample cases. First, though, we must describe Alice's concept formation in greater detail.

5 Marking Salient Attributes

Alice's concept formation uses the Marking Salient Attributes (MSA) technique. Alice begins by eating and recording the utility of a number of mushrooms. Let's assume her first mushroom is nourishing. As soon as she eats a discordant mushroom (i.e., a harmful one), she marks all of the attributes of this new mushroom that distinguish it from the nourishing ones. As each mushroom is added these distinguishing marks are added. For example,

```
mushroom...............1.......2.......3
color (red blue).......r(3)....r(3)....b(1 2)
texture (spots plain)..s(3)....p.......p(1)
shape (circ irr).......c.......c.......c
utility................+1......+1......-1
```

The numbers in the parentheses indicate which discordant mushrooms this attribute distinguishes. Thus, the texture mark indicates the attribute value *plain* distinguishes discordant mushroom 3 from 1, while color distinguishes mushroom 3 from both 1 and 2.

Clearly, as Alice encounters more and more mushrooms some simplification is necessary. This is done by two techniques: reduction and combination. After Alice has encountered sufficiently many mushrooms (including some discordant ones) she attempts a reduction. For example, mushroom 3 has two attributes that distinguish it from mushroom 1—but only one is necessary. The rule Alice employs is to retain the mark for the longer list. This serves to retain as salient those attributes which have the most discriminatory power. Thus the mark (1) will be removed from attribute value *plain* of 3 and so too will its corresponding mark on mushroom 1 yielding:

```
mushroom...............1.......2.......3
color (red blue).......r(3)....r(3)....b(1 2)
texture (spots plain)..s.......p.......p
shape (circ irr).......c.......c.......c
utility................+1......+1......-1
```

Having reduced the total set of marks to a salient set, Alice can now combine nodes. A mushroom node will combine with a second concordant node (i.e., they will be classed together) if the second node has all the same salient attributes as the first node. In the above case mushrooms 1 and 2 combine, producing 1*:

```
mushroom...............1*(1 2).....3
color (red blue).......r(3)........b(1 2)
texture (spots plain)..-...........p
shape (circ irr).......c...........c
utility................+1..........-1
```

where "–" means 'don't care'; shape is retained for 1* as a non-salient attribute in common; while utility is *always* retained—for it is the driving force behind all categorization.

This technique of course loses information—namely the 'don't care' attributes of combined nodes. But losing what is apparently irrelevant information is, in fact, the main goal of all methods of abstraction.

Exception specialization can be seen when mushroom 4 arrives, which, after marking and reduction, gives us:

```
mushroom...............1*(1 2)....3........4
color (red blue).......r(3).......b(1).....r
texture (spots plain)..-..........p........p
shape (circ irr).......c(4).......c........i(1)
utility................+1.........-1.......-1
```

Alice expects mushroom 4 to be nourishing as it matches all the salient attributes of 1. She is surprised when she eats it and finds it is harmful. To deal with this she must find an attribute that was previously non-salient, but may be made salient as a newly distinguishing feature. The marking technique finds this attribute (shape) and marks it. Now two attributes are salient for concepts 1* and 4—and red, irregular mushrooms are kept distinct (as a separate concept) from red, circular mushrooms. Alice proceeds eating mushrooms and filling in their salient features. Three things can happen to a new mushroom: It may match some existing concept and become (and remain) incorporated within it; it may match an existing concept, but end up requiring exception specialization; or, it may be added to the conceptual hierarchy as a new concept in its own right, with the possibility of subsequent combination.

6 Comparative Results

MSA was run initially on some simple test cases to compare its approach with classical concept formation. An example is the following test case of five mushrooms:

	green	blue	red	yellow
circ	-	N2	-	-
oblong	-	-	N3	-
irr	P1	+	P4	P5

The sequence of mushrooms is indicated as Pn and Nn, reflecting whether the nth mushroom produced a positive or negative change in utility. The +/- signs indicate what Alice is predicting after having sampled all five mushrooms. Alice has combined mushrooms (3 1) (3 3) and (3 4) [identified by (row column)] into a single concept with the shape attribute marked as salient, since shape is all that is needed to distinguish the negative from positive instances here. In effect, Alice is operating with three concepts at this point—splitting on the shape of mushrooms. This is, of course, similar to what we get with version space methods and ID3. However, adding a sixth test mushroom we get:

	green	blue	red	yellow
circ	-	N2	-	-
oblong		P6	N3	
irr	P1	+	P4	P5

Mushroom 6 forces the color attribute into salience together with the shape attribute vis-a-vis the third and sixth mushroom. Since mushrooms in positions (2 1) and (2 4) do not match either of these on both attributes, they do not match any of the current mushroom concepts; therefore, Alice does not predict whether they are beneficial or harmful. Ordinary version space methods break down altogether with such examples, since the positive concept can only be a *disjunction* of convex regions; however, they can be modified to discover subconcepts by splitting the training data so as to allocate positive instances to convex regions only. In that case, they may produce predictions for the unexamined instances, classifying (2 1) as positive and (2 4) as negative, for example.

This example highlights a major difference between Alice's concept formation and traditional methods: namely, Alice's conservativeness. Alice certainly generalizes positive instances into classes that cover unexamined cases (exhibiting inductive generalization), but she is nonetheless far less prone to over-generalization than the competing algorithms. We view this as a potential competitive advantage in dangerous environments; it is no good considering green oblong mushrooms, which have not been sampled, to be aś safe as blue oblong mushrooms, which have been.

We have extended our experimental results to include some of the static test cases from the UC Irvine machine learning database. Those cases are mostly designed to test the optimal performance of a learning algorithm, measured by *accuracy*—i.e., the percentage of correct predictions. Since it is an important feature of Alice that she can refrain from prediction when that is prudent, we prefer to measure her performance in terms of *reliability*—i.e., 100% minus the percentage of failed predictions. For competitive learning algorithms reliability and accuracy are identical. Thus far, we have obtained these results:[1]

Test	MSA/R	MSA/A	AQR	ID3	C4.5 Trees	C4.5 Rules
Monks 1	99.1	96.3	95.9	98.6	75.7	100
Monks 2	74.3	65.5	79.6	67.9	65.0	65.3
Monks 3	94.4	88.9	87.0	94.4	97.2	96.3
Mushrooms	99.2	95.9	--	--	98.3	98.5

Although Alice was not designed with such static environments in mind, she has done quite a creditable job here. She has in all but one case outperformed her competitors on most cases, when measured by reliability; and she remains competitive when measured by accuracy. Thus, we suggest that in many realistic environments—where an unwarranted prediction can be much worse than no prediction—Alice will outperform alternative methods.

Interpretation

Alice is by her nature capable of forming concepts of arbitrary geometric complexity in the instance space—although her conceptual structure may become uneconomic if too many attributes are forced into salience. In this flexibility, and in the naturalness of her formation rules, she appears to offer some advantage over competing paradigms for

[1] MSA/R is the reliability score, while MSA/A is accuracy. AQR is one of the AQnn family of version-space methods; ID3 and C4.5 are Quinlan's information-theoretic algorithms for building classification trees. Results for AQR are from Kreuziger, et al. (1991); results for ID3 are from the *update* file at the UC Irvine database; the Monks results for C4.5 are from Cameron-Jones and Quinlan (1993); the mushroom test results for C4.5 are from Van Horn and Martinez (1993). In order to have comparable results we have used Van Horn and Martinez's training regime for Alice in the mushroom test—i.e., we have used 200 randomly selected training cases and 7,924 test cases. For all tests we turned off Alice's learning ability after the training cases.

concept formation. Alice shows a praise-worthy restraint in her inductive generalizations implicit in concept formation, yet she is not so restrained that she cannot achieve quite good accuracy scores on standard tests.

Alice's conservativeness, of course, *can* also be a disadvantage when the environment demands a prediction—that is, an action that is best based upon a prediction which Alice is reluctant to give. For example, suppose she is near expiration due to hunger and must take a chance on one or another of two mushrooms without predicted values. A natural approach for dealing with such circumstances is to extend Alice's predictive abilities to incorporate a distance metric in the attribute space. Alice could then either select the class closest to the unknown mushroom or use a weighted average of nearby classes for making her prediction.

7 Future Work

This paper reports on the very early stages of research on primitive concept formation and primitive induction. The next experimental step is to examine the performance of Alice in a rich testing environment (RALPH) developed at UC Berkeley for testing machine learning programs (cf. desJardins, 1992). That environment allows for noisy domains and time-constrained problem solving. We shall simultaneously test alternative machine learning algorithms under like constraints (including some we haven't yet examined experimentally, such as Oliver, Dowe, and Wallace, 1992). We also plan to continue using problems from the UC Irvine archive; however, our goal will not be to optimize performance without regard for pragmatics, as is usual, but simply to understand the limits and potential for Alice's conceptualizations in complex cases.

As she stands, Alice already incorporates a fair number of the desiderata we've listed for autonomy. She is able to develop a conceptual hierarchy in primitive circumstances, generalizing on positive examples and specializing on negative examples. Her conceptual hierarchy is developed in a manner responsive to her current goals, and, as the set of goals she is capable of entertaining is expanded, we expect her learning methods to continue to reflect them. Her learning regime is explicitly incremental and the representations learned ready to be deployed at any time necessary. Furthermore, her conceptualizations and concomitant predictions are explicitly defeasible—indeed, Alice is constantly revising her old judgments by way of exception specialization. Alice's most notable current deficiency is her inability to deal with a stochastic environment. We expect the incorporation of frequency information and its use in concept formation and prediction to be one of the extensions we will incorporate in the near future (see Korb and Dowe, forthcoming, for a theoretical treatment of frequency-based induction).

8 Conclusion

Cognitively plausible methods of concept formation (and, presumably, learning in general) show promise for supporting learning and reasoning in uncertain, pragmatically constrained and dynamically changing environments. While the classical supervised techniques are effective in the static environments for which they were developed, we speculate that—and have some modest evidence that—they will be far less effective than Alice in more demanding and less predictable problem domains.

References

Brooks, R. (1991) 'Intelligence without Representation,' *Artificial Intelligence 47*: 139-159.

Cameron-Jones, R.M. and Quinlan, J.R. (1993) 'First Order Learning, Zeroth Order Data,' in C. Rowles, H. Liu, and N. Foo (eds.) *AI '93* (pp. 316-321). World Scientific.

desJardins, M. (1992) *PAGODA: A Model for Autonomous Learning in Probabilistic Domains.* Report UCB/CSD 92/678, Computer Science Division, EECS, UC Berkeley.

Holland, J.H. and Holyoak, K.J. and Nisbett, R.E and Thagard, P.R. (1986) *Induction.* MIT.

Korb, K.B. (forthcoming) 'Inductive Learning and Defeasible Inference,' in *The Journal of Experimental and Theoretical Artificial Intelligence.*

Korb, K.B. and Dowe, D.L. (forthcoming) 'The ROACH Model of Induction: Explorations in Primitive Intelligence,' in preparation.

Kreuziger, J., Hamann, R. and Wenzel, W. (1991) 'Comparison of Inductive Learning Programs,' in *The MONK's Problems: A Performance Comparison of Different Learning Algorithms*, technical report CMU-CS-91-197, pp. 59-80.

Mitchell, T. (1977) 'Version Spaces: A Candidate Elimination Approach to Rule Learning,' *Proceedings of the Fifth International Joint Conference on Artificial Intelligence.*

Oliver, J., Dowe, D. and Wallace, C.S. (1992) 'Inferring Decision Graphs Using the Minimum Message Length Principle,' *Proceedings of the 1992 Australian Artificial Intelligence Conference.*

Ram, A. and Hunter, L. (1992) 'The Use of Explicit Goals for Knowledge to Guide Inference and Learning,' *Applied Intelligence, 2*: 47-73.

Quinlan, J. (1983) 'Learning Efficient Classification Procedures and their Application to Chess End Games,' in R. Michalski, J. Carbonell and T. Mitchell (eds.) *Machine Learning: An Artificial Intelligence Approach.* Palo Alto: Tioga.

Van Horn, K.S. and Martinez, T.R. (1993) 'The BBG Rule Induction Algorithm,' in C. Rowles, H. Liu, and N. Foo (eds.) *AI '93* (pp. 348-355). World Scientific.