Categorization, Concept Learning, and Problem Solving: A Unifying View

Douglas Fisher
Department of Computer Science
Vanderbilt University
Nashville, TN 37235

Jungsoon Yoo
Department of Computer Science
Middle Tennessee State University
Murfreesboro, TN 37132

Abstract: This paper illustrates the synergism between categorization and problem solving, and between artificial intelligence and cognitive psychology, with two computational models. The first, COBWEB, is a learning system for hierarchical categorization that provides good fits to experimental data on basic level, typicality, and fan effects. The second model, EXOR, learns to solve problems more effectively by extending the basic COBWEB learning and categorization strategies in a direction that is cognizant of background knowledge or preconceptions on the part of the learner. Our adaptation illustrates the importance of categorization in problem solving. It also promotes a unique perspective that unifies fan effects with typicality and basic-level phenomena, and identifies a phenomenon, basic levels of problem solving, which appears to be novel in the literature and speaks to issues of learning and training of problem-solving skills in humans and machines.

Our methodological viewpoint agrees with Anderson’s ideas of rational analysis: if humans are bounded-rational agents, then a reasonable starting point for modeling their behavior is an objective function that describes ideal behavior conditioned on resource (e.g., memory) constraints. This view is consistent with traditional research at the interface of artificial intelligence and cognitive psychology, but we stress throughout that to be maximally useful, computational models should move beyond the known experimental data, thus providing guidance for further psychological study.

Contents

1 Introduction 1

2 Concept Learning and Categorization 2
  2.1 COBWEB: A Concept Learning System 2
  2.2 A Psychological Perspective 4
    2.2.1 Basic Level Effects 5
    2.2.2 Typicality Effects 6
    2.2.3 Fan Effects 6
  2.3 Summary 7

3 Categorization and Problem Solving 8
  3.1 Theory and Data 9
  3.2 A Computational Perspective on Problem Solving 10
    3.2.1 Problem Solving as Search 10
    3.2.2 Improving the Efficiency of Problem Solving 12
  3.3 EXOR: Improving Problem Solving by Concept Learning 14
    3.3.1 An Example of Problem Solving by Categorization 14
    3.3.2 Categorization and Learning 15
    3.3.3 Empirical Results 17
    3.3.4 Inferred Features and Pruning 18
  3.4 Summary 20

4 General Discussion 21
  4.1 Categorization and Inference 21
  4.2 Learning Issues 22
  4.3 Abstraction and Basic Levels of Problem Solving 23
  4.4 Prescriptions for Learning and Training 25

5 Concluding Remarks 25

Acknowledgements 27

Endnotes 28

References 29

Index terms 34
1 Introduction

Categorization and problem solving are widely studied in psychology and artificial intelligence (AI), but these areas of research have been largely segregated. Several chapters in this volume seek to rectify this situation by discussing relationships between categorization and problem solving and the memory structures that support them (Nakamura, this volume; Brewer, this volume). We address these issues as well, albeit with considerable attention to learning issues as well. In particular, this chapter views categorization as an integral aspect of problem solving: schemas are organized by well-known principles of category structure, problems are categorized in a manner that suggests schemas appropriate to their solution, and the transition from novice to expert problem solving can be understood as largely a process of concept learning.

Our discussion draws from AI and cognitive psychology, which have followed parallel paths on a number of issues. Notably, both AI methods and psychological models of categorization and concept learning have explored the utility of probabilistic concept representations (Smith & Medin, 1981; Fisher & Pazzani, 1991; McDonald, this volume; Malt, this volume; Bareiss & Slator, this volume; Taraban & Palacios, this volume), and the relative roles of prior knowledge and observational data on concept learning and categorization (Wisniewski & Medin, 1991; Mooney, this volume; Mumma, this volume; Murphy, this volume; Ward, this volume). In general, psychology and AI have have much to say to one another. The common wisdom is that psychology provides a ‘specification’ of intelligence, and AI provides tools for implementing this specification. However, the relationship should go beyond this loose coupling of disciplines. AI models of human intelligence should make predictions about currently unexplored or debated phenomena, thus suggesting areas for further psychological study. Without explicit attention to this feedback loop, the interaction between AI and psychology is of debatable utility.

This chapter illustrates the synergism between categorization and problem solving, and between psychology and AI, using a computer model called EXOR (Yoo & Fisher, 1991a,b). This system categorizes a problem using a ‘knowledge base’ of previously-experienced problems and their solutions. The solutions to problems found through categorization are reused in the solution to the new problem. Once solved, this new problem and its solution are added to the memory of experiences for future use. The principles underlying EXOR derive from a computer-implemented model of categorization called COBWEB (Fisher, 1987b), which traces its ancestry to EPAM (Feigenbaum, 1961) and other hierarchical categorization models (Kolodner, 1983). We begin by briefly describing COBWEB to better convey its principles as realized in EXOR, and to highlight the commonality and differences between categorization models of object memory and problem solving more generally.

2 Concept Learning and Categorization

Concept learning is a major area of study in machine learning and cognitive psychology. The concept learning task requires a learner to discover concepts (i.e., rules) that adequately describe observations. For example, the learner may learn rules that distinguish
patients with hypothyroid, hyperthyroid, and other thyroid conditions from patients that are healthy. This is an example of a supervised learning task, in which observations are labeled by category names; the learner's task is to characterize the commonalities between members of the same category and differences between contrast categories.

In contrast, an unsupervised task requires the learner to discover category structure in initially unclassified data. Experimental tasks like sorting are unsupervised in this sense, and a number of psychological models and computer implementations have been developed. EPAM (Feigenbaum, 1961; Richman, 1991) is an early computer-implemented cognitive model of unsupervised learning. It assumes that a hierarchical memory is gradually constructed by finding features that discriminate new observations from previous ones. EPAM is a learning model of human object recognition, which identifies objects that have been previously observed. However, this recognition process is mediated by generalized categories that are represented by nodes in an evolving categorization hierarchy. A more recent model of unsupervised learning has been developed by Anderson (Anderson, 1990; Anderson & Matessa, 1991), which attempts to discover the 'most probable' categories in the data using a Bayesian probability approach.

EPAM and Anderson's approach differ in many ways, but both gradually construct categorization trees over a stream of unclassified observations. This process has been termed concept formation and many other models in this paradigm have been developed (Fisher & Pazzani, 1991). This section describes one such system, COBWEB (Fisher, 1987b).

2.1 COBWEB: A Concept Learning System

Unsupervised tasks require the learner to find categories in data. However, there are a vast number of ways to group observations into contrast categories. Clearly, most of these possibilities are not informative to a learner. To ferret out the informative categorizations an unsupervised system relies on a measure of category quality, and a method of searching through the possibilities for the categorization that optimizes (or is satisfactory by) this measure. Corter and Gluck (1992) suggest that certain categories are preferred because they best facilitate predictions about observations in the environment. If observations are represented as sets of features, \( V_j \), then Corter and Gluck's measure of category utility can be partially described as a tradeoff between the expected number of features that can be correctly predicted about a member of a category \( C_k \) and the proportion of the environment, \( P(C_k) \), to which those predictions apply:

\[
P(C_k)E(\# \text{ of correctly predicted } V_j | C_k).
\]

For example, little can be predicted about a highly-general category like 'animals', but those features that can be predicted (e.g., 'animate') apply to a large population. In contrast, many features can be predicted with near certainty about highly-specific categories like 'robins', but these predictions are true of a relatively small population. Intuitively, a category of intermediate generality such as 'birds' maximizes the tradeoff between the expected number of accurate predictions, and the scope of their application.

The expectation can be further formalized by assuming that predictions will be generated by a probability matching strategy: one can predict a feature with probability
$P(V_j|C_k)$, and this prediction will be correct with the same probability. Thus,

$$E(\# \text{ of correctly predicted } V_j|C_k) = \sum_j P(V_j|C_k)^2.$$  

Gluck & Corder (1985) define category utility as the increase in the expected number of features that can be correctly predicted, given knowledge of a category, over the expected number of correct predictions without such knowledge:

$$CU(C_k) = P(C_k)[\sum_j P(V_j|C_k)^2 - \sum_j P(V_j)^2].$$

Note that by moving the $P(C_k)$ term into the summation and applying Bayes rule

$$P(C_k)\sum_j P(V_j|C_k)^2 = \sum_j P(V_j)P(C_k|V_j)P(V_j|C_k).$$

Thus, $CU(C_k)$ is also a tradeoff between standard measures of cue validity, $P(C_k|V_j)$, and category validity, $P(V_j|C_k)$, weighted by the probability of $V_j$. As Medin (1983) notes, cue validity will tend to be higher for very general categories, since they have fewer contrasts. Inversely, category validity will tend to be higher for highly specific categories.

A concept formation system called COBWEB (Fisher, 1987b) uses category utility to partition a set of observations into contrasting categories, $C_k$, so as to maximize the average utility of categories in the partition,

$$\frac{\sum_{k=1}^n CU(C_k)}{n},$$

where $n$ is the number of categories in the partition. Because category utility requires only information about individual feature distributions within each $C_k$, one can effectively represent a category with a probabilistic or independent cue representation (Smith & Medin, 1981), where each feature, $V_j$, is weighted by $P(V_j|C_k)$.

Figure 1 illustrates how COBWEB organizes categories at multiple levels of abstraction. In this example, observations correspond to animal descriptions. Each category is also weighted by the proportion of observations, $P(C_k)$, classified under it.

The method that COBWEB uses to incorporate a new observation into an existing categorization tree is designed for simplicity and efficiency. COBWEB assimilates an observation by evaluating the partitions that result by adding the observation to each existing category, and the partition that results from creating a new singleton category. Intuitively, this latter action occurs if the observation is sufficiently dissimilar from each of the existing categories. It then evaluates each of these alternatives using category utility and retains the best choice. Incorporating a new animal description into the tree of Figure 1 would require three evaluations at the top level of the tree: adding the instance to the 'vertebrate' category, adding it to the 'invertebrate' category, and creating a new category that contained only the new instance. If the instance is incorporated into an existing category, then the observation is assimilated into the respective subtree by the same procedure. Anderson and Matessa (1991) describe an approach in which a Bayesian measure guides object assimilation in the same manner.

We noted that category utility and COBWEB were motivated by a desire to facilitate inference – in this case, the prediction of features that are not directly observed in a
new object. As with assimilation, COBWEB uses category utility to guide object categorization: an observation is sorted down a path of 'best matching' categories to a point where the unknown feature is best predicted. Some features may be best predicted at the leaves (specific, past instances) of the categorization tree, while others may be best predicted at categories of intermediate generality (Fisher, 1989). For example, consider a zoological categorization that decomposes ‘animals’ into ‘vertebrates’ and ‘invertebrates’, and then further decomposes these categories into ‘mammals’, ‘birds’, etc. If we observe that an observation has ‘feathers’, then the path of best matching categories will include ‘animals’, ‘vertebrates’, and ‘birds’. During top-down categorization we may choose to predict that the object is probably ‘animate’ when categorization reaches the ‘animal’ category, that it has a ‘backbone’ when categorization reaches the ‘vertebrate’ category, and that it probably ‘flies’ at the ‘bird’ category. Thus, to the extent possible, inferences about an incomplete observation are accumulated as one descends the categorization hierarchy. This ability improves with learning. The exact nature of the learning curve and best point of prediction varies between features, and is based on the degree that the features are intercorrelated with other features (Fisher & Langley, 1990).

2.2 A Psychological Perspective

The analysis above is primarily computational: COBWEB improves inference as learning proceeds, and it does so in a relatively efficient manner. However, these concerns
with accuracy and efficiency are intricately linked to psychological concerns. For example, during top-down categorization COBWEB places an observation in one category at each level of the categorization tree; once made, it never reconsiders a categorization in light of subsequent data. This 'localized' categorization policy is space and time efficient, but it also renders the system sensitive to the order of incoming data (Fisher, Xu, & Zard, 1992), which also characterizes aspects of human learning (Kline, 1983).

More generally, COBWEB's development is consistent with a rational analysis (Anderson, 1990): a reasonable starting point for modeling human processing is one that is optimal given the resource limitations under which humans operate. Gluck and Corter (1985) motivate category utility in precisely this way. Rationally, we wish to increase the expected number of correct inferences, $P(V_j|C_k)^2 - P(V_j)^2$, that can be made about an observation, but this aspect of category utility will be maximized for the most specific categories possible. Unfortunately, while a computer may easily memorize specific instances and treat them as singleton categories, humans often cannot do so. Thus, category utility also rewards more general categories that represent some form of data compression through the $P(C_k)$ term. Consistent with a bounded-rational view of humans (Simon, 1969), category utility trades off accuracy with space efficiency. As we noted, category utility can be viewed as a tradeoff between cue and category validity, which also explains a preference for categories that are not overly general or specific as well (Medin, 1983).

2.2.1 Basic Level Effects

Psychological studies have shown that within hierarchical classification schemes there appears to be a basic level preferred by human subjects (Rosch, Mervis, Gray, Johnson, & Boyes-Braem, 1976; Lassaline, Wisniewski, & Medin, in press). For example, in a hierarchy containing {animal, vertebrate, mammal, dog, collie}, subject behavior may indicate that 'dog' lies at the basic level. Corter and Gluck developed category utility as a predictor of these basic categories in humans.

Basic level studies by Murphy and Smith (1982) trained subjects to categorize artificial tools relative to hierarchically-organized categories. After training, subjects more quickly affirmed that objects were members of their respective (intermediate) basic categories, than they confirmed membership relative to superordinate or subordinate categories. Table 1 shows the predicted results of COBWEB's classification strategy on these data. The 'True cases' measured subject's response time to confirm that stimuli (observations) were members of a given superordinate, basic, or subordinate category. Predicted response times are computed through a linear regression ($r = -0.94$) over a $CU$-based measure of similarity between objects and learned categories (Fisher & Langley, 1990). Murphy and Smith also tested 'False cases', which indicated that subjects were faster at recognizing that a stimulus was not a member of a basic category, than disconfirming membership relative to the other two levels. Three data points per case are too few for any conclusions about quantitative fit, but qualitatively the model ranks categories in the same order as response time ($r = -0.96$). This includes a predicted preference for subordinate over superordinate levels.

This account follows directly from category utility's tradeoff between accuracy and space efficiency, which tends to favor categories at intermediate levels of abstraction.
Table 1: Human and predicted response times for the Murphy and Smith (1982) data. Adapted from Fisher and Langley (1990).

<table>
<thead>
<tr>
<th></th>
<th>True Cases</th>
<th></th>
<th>False Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Response time (msec)</td>
<td>Predicted time</td>
<td>Response time (msec)</td>
</tr>
<tr>
<td>Superordinate</td>
<td>879</td>
<td>869</td>
<td>882</td>
</tr>
<tr>
<td>Basic</td>
<td>678</td>
<td>646</td>
<td>714</td>
</tr>
<tr>
<td>Subordinate</td>
<td>723</td>
<td>764</td>
<td>691</td>
</tr>
</tbody>
</table>

This result is not surprising since Corter and Gluck (1992) developed the measure as a predictor of basic levels, though we have extended their analysis to ‘False’ cases.

2.2.2 Typicality Effects

In addition to basic levels, COBWEB has been applied to data on typicality effects from Rosch and Mervis (1975). For example, subjects will more quickly affirm that a ‘robin’ is a ‘bird’ than they will affirm that a ‘chicken’ is a ‘bird’. The relative ranking of test items corresponds to a typicality ranking of category members.

Rosch and Mervis found that category members sharing features with many other members of the same category tend to be judged more typical. In addition, when a disjoint, contrasting category is involved, members that share few features with members of the contrasting category tend to be judged more typical. This sensitivity to intra-category and inter-category overlap of features is captured by the notion of family resemblance. Rosch and Mervis trained subjects to recognize and distinguish nonsense strings from two contrast categories. In experiments testing the effect of intra-category similarity, members of one category were JXPHM, QBLFS, XPHMQ, MQBLF, PH-MQB, and HMQBL. These strings shared no features (symbols) in common with members of the contrast category, but they varied in their intra-category overlap. For example, the symbols of JXPHM are present in an average of 2.0 other strings of category A. This string is categorized slowly, and thus judged to have low typicality. In contrast, the symbols of HMQBL are present in an average of 3.2 other strings of category A, yielding faster response and high typicality judgments. Similar experiments tested variability of inter-category overlap: response time increases and typicality decreases with greater inter-category overlap.

Actual response times and those predicted by the COBWEB model are shown in Table 2. The good quantitative fits $[F(1, 4) = 114.1, p < 0.001]$ are due to category utility’s tradeoff between cue validity and category validity, which accurately reflect differences in inter-category overlap and intra-category overlap, respectively.

2.2.3 Fan Effects

Finally, COBWEB’s classification schemes fit data on fan effects by Anderson (1974), in which subjects were asked to recognize whether they had seen a test sentence or
Table 2: Human and predicted response times for Rosch and Mervis (1975) data. Adapted from Fisher and Langley (1990).

<table>
<thead>
<tr>
<th></th>
<th>Response Time (msec)</th>
<th>Predicted Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intraoverlap</td>
<td></td>
<td></td>
</tr>
<tr>
<td>high</td>
<td>560</td>
<td>535</td>
</tr>
<tr>
<td>med</td>
<td>617</td>
<td>615</td>
</tr>
<tr>
<td>low</td>
<td>692</td>
<td>713</td>
</tr>
<tr>
<td>Interoverlap</td>
<td></td>
<td></td>
</tr>
<tr>
<td>low</td>
<td>909</td>
<td>968</td>
</tr>
<tr>
<td>med</td>
<td>986</td>
<td>995</td>
</tr>
<tr>
<td>high</td>
<td>1125</td>
<td>1062</td>
</tr>
</tbody>
</table>

not during previous training (e.g., ‘The teacher is in the bank’, ‘The fireman is in the park’). Each sentence contains a ‘subject’ (e.g., ‘teacher’, ‘doctor’) and a ‘location’ (e.g., ‘church’, ‘bank’). Anderson’s findings indicate that the greater the featural overlap of a test sentence with training sentences, then the longer the time required to confirm (True cases) or disconfirm (False cases) its presence in earlier training. Silber and Fisher (1989) and Fisher and Langley (1990) viewed COBWEB’s account of this data as a special case of typicality. The task was to match a test sentence against a past sentence, which can be regarded as a singleton category. Because there is only one object per category, the intra-category overlap in training sentences is constant. Thus, differences in response time are due exclusively to inter-category overlap – the overlap between sentences. Actual response times and predicted response times for the ‘True’ cases \(F(1, 7) = 36.3, p < 0.001\) and ‘False’ cases \(F(1, 7) = 17.1, p < 0.004\) are shown in Table 3. Generally speaking, our analysis considers the memory process of recognition as a special case of categorization, whereas Anderson (1990) and psychological accounts generally, consider them distinct, though probably related.

In addition, this general view also speaks to the negative fan effect (Reder & Ross, 1983), which indicates that the more that one knows about a particular observation relative to other observations, then the faster it will be recognized on average. In this case, we are increasing the intra-category similarity of one singleton category relative to others, since there are a greater number of facts known about it. Again, this is a degenerate case of typicality; a robust model of typicality appears sufficient to largely explain the phenomena.

2.3 Summary

This section has briefly described the COBWEB system, a model of categorization and concept learning. Inferences about unobserved dimensions of an object are accumulated
Table 3: Human (msec) and predicted (in parentheses) response times for Anderson’s (1974) fan effect data. Adapted from Fisher and Langley (1990).

<table>
<thead>
<tr>
<th>True cases</th>
<th>False cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject overlap</td>
<td>subject overlap</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1111</td>
</tr>
<tr>
<td></td>
<td>(1120)</td>
</tr>
<tr>
<td>location overlap 2</td>
<td>1167</td>
</tr>
<tr>
<td></td>
<td>(1157)</td>
</tr>
<tr>
<td>3</td>
<td>1153</td>
</tr>
<tr>
<td></td>
<td>(1184)</td>
</tr>
</tbody>
</table>

during categorization, and learning improves the prediction accuracy of this process. The system has also been evaluated as a parameter-free model of human categorization. It is not a process model in the usual sense, since we do not suggest how the CU-based categorization procedure might be ‘implemented’ by lower-level cognitive mechanisms. However, indexing schemes closer to the process level have been proposed (Fisher & Langley, 1990).

The model accounts for basic level and typicality effects, as well as interactions between them (Jolicoeur, Gluck, & Kosslyn, 1984; Fisher & Langley, 1990). However, the model is particularly novel in its account of fan effects, which are not generally treated in a categorization mold (Anderson, 1990). This highlights an important principle of cognitive modeling: to be useful, a cognitive model should go beyond the known data, thus suggesting new conceptual frameworks and experimental directions.

COBWEB’s approach to categorization and learning shapes our view of problem solving, but inferences correspond to partial solutions that are accumulated into a complete solution as a problem is categorized relative to past experiences.

3 Categorization and Problem Solving

It is apparent that human problem solving typically involves some form of categorization. For example, in chess (DeGroot, 1966), experts rapidly categorize board patterns and act on these categorizations during play. A classic study by Chi, Feltovich, and Glaser (1981) found that experts and novices alike classify physics problems, and (at least in the case of experts) these categorizations determine starting points for problem solving. Production-system models (Anderson & Kline, 1979; Simon & Lea, 1979) make the ‘problem solving as categorization’ view explicit. Categorization leads to the application of ‘matching’ operators, thus generating inferences or actual changes to the environment.

Given the apparent importance of categorization, it is natural to look to traditional models of the process, including hierarchical strategies like COBWEB. However,
COBWEB’s representation of concepts needs to be revised to handle problem-solving tasks.

3.1 Theory and Data

Traditionally, research in both machine and human learning has focused on surface features as the basis for concept representations. Within psychology and machine learning there is growing dissatisfaction with this simple featural model (Medin, 1989; Porter, Bareiss, & Holte, 1990). In particular, background knowledge – what the learner already knows and which is not perceivable in the data – appears to play a vital role in category formation. For example, consider the task of teaching subjects about straight card hands in poker. Two sample straights include ‘((5, clubs), (6, diamonds), (7, diamonds), (8, hearts), (9, clubs))’ and ‘((7, spades), (8, spades), (9, clubs), (10, hearts), (Jack, diamonds))’. To correctly learn the concept requires knowledge that (3 succeeds 2), (4 succeeds 3), .. (Jack succeeds 10), .. (Ace succeeds King), which is not an explicit part of the observations, but must be brought to the task from the learner’s background knowledge (Vere, 1978).

In problem solving the role of background knowledge is particularly important. In fact, all those aspects of a problem’s solution that are not an explicit part of the problem statement are background knowledge. Studies by Chi et al. (1981) illustrated the importance of background knowledge to human categorization and problem solving. They required subjects to sort physics problems into categories of their own design, thus revealing significant differences between the criteria used by ‘experts’ (i.e., graduate students in physics) and ‘novices’ (i.e., undergraduates) to create categories. Novices sorted based on surface features – those that were referred to in the problem statement (e.g., reference to an ‘inclined plane’, ‘friction’, etc.), while experts formed categories of problems that required the application of similar solution strategies (e.g., application of ‘Newton’s second law’). In addition, experts generally required more time to produce an initial sort than novices. These findings indicate that expert subjects make an initial ‘qualitative analysis’ of a problem prior to classification, and derive abstract features from background knowledge (e.g., laws of physics).

Chi et al’s (1981) study suggests that experts use abstract features, as opposed to surface features, to define categories in problem-solving contexts. However, in many cases, surface features more directly guide categorization, even in experts, when they are well-correlated with the abstract features relevant to a problem’s solution (Ross & Spalding, 1991). For example, observing ‘boat’ in the description of a simple algebra story problem will often cue the reader (perhaps incorrectly) that it is a motion problem that involves time, rate, and/or distance calculations.

These examples seem to offer two extreme views of categorization as being directed predominantly by abstract or surface features. However, as Ross and Spalding recognize, these extremes are oversimplified. For example, Wisniewski and Medin (1991) found that surface and background interact in subtle ways and continuously throughout categorization. Their studies looked at how adult subjects categorized children’s drawings of people. When told that a drawing was by a mentally-gifted child (which may or may not actually have been the case), subjects might postulate an abstract feature such as ‘draws well’. The presence of ‘buttons’ (i.e., a surface feature) in the drawing
might confirm ‘detail’ (i.e., an abstract feature) in the drawing, which then might be used to confirm the initially hypothesized characteristic of good drawing. Failure to find confirming surface features would cause some revision in the subject’s hypothesis and interpretation of the drawing. In general, they argue that categorization iterates between reliance on surface and abstract features.

Despite the apparent interaction between surface and background knowledge in categorization and problem solving, COBWEB and other ‘similarity-based’ categorization models typically make no distinction between these forms of knowledge, and thus appear limited as models of categorization, particularly as it relates to problem solving.

3.2 A Computational Perspective on Problem Solving

This section describes a categorization model of problem solving inspired by research on COBWEB, but cognizant of background knowledge. We open with a view of problem solving as a search-intensive process, then move to issues of learning that render the task more efficient.

3.2.1 Problem Solving as Search

Traditionally, much of the research on problem solving in AI and cognitive psychology view it as a search task (Simon, 1969). Given a problem description, the task is to search background knowledge for a solution that satisfies all the constraints in the problem’s description. As an illustrative example, consider an algebra story problem like those cataloged by Mayer (1981):

“A train leaves a station and travels east at 72 km/h. Three hours later a second train leaves and travels east at 120 km/h. How long will it take to overtake the first train?”

We assume that a (human or machine) problem solver’s background knowledge includes domain-specific knowledge about motion problems (e.g., \( D = R \times T \)) and general algebraic knowledge (e.g., \( a \times b = c \rightarrow a = c/b \)). Problem solving requires a search of the possible ways to chain these background rules together in order to derive a complete solution. However, there are many factors that can complicate the process. In some cases the problem solver’s knowledge may be incorrect or incomplete, and the problem solver may or may not have access to external assistance. For example, the problem solver may realize that the time, \( T_2 \), traveled by the second train equals the distance that it has traveled divided by its rate (\( T_2 = D_2/120 \)), thus triggering an attempt to derive \( D_2 \) from the known facts: \( D_2 = R_2 \times T_2 = 120 \times T_2 \). However, this circularity leads to a ‘dead end’.

Alternatively, the problem solver might observe that the second train travels three hours less than the first train, or \( T_2 = T_1 - 3 \). Thus, solving \( T_1 \) provides a solution to \( T_2 \). In general, many paths may be tried before a solution is found. The line of reasoning above will be successful if the problem solver observes that the distances traveled by the two trains to point of overtake are equal, \( D_1 = D_2 \). This additional constraint is necessary to solve the problem. If the problem solver fails to encode this knowledge from the problem statement or lacks background knowledge to exploit it, then this line of reasoning will fail.
Figure 2: A specific solution and generalized schema for OVERTAKE.

An alternative solution to the overtake problem is abbreviated in Figure 2(a). The solution is expressed in a formal notation required by the computer, but intuitively it encodes that the time until overtake ($T = \Delta t/R$) is obtained from the distance that must be made up by the faster train, where $D1$ is the distance traveled by the first train before the second train starts, and the relative rate of travel of the second train ($R = R2 - R1$). Again, if appropriate knowledge is lacking or incorrect, then this line of reasoning will fail as well.

The search for a correct solution to a problem will succeed if the necessary constraints and facts are encoded from the problem statement and the required background knowledge is present. In cases where problem solving is being learned, simultaneous fulfillment of these conditions is often not realizable. Rather, external coaching is required by an instructor. Nonetheless, the computer model that we will describe assumes that any problem presented to the system can be solved correctly and to completion. In this case, learning is exclusively concerned with speeding up problem solving. Initially, even with complete and correct background knowledge, the problem solver may search in a relatively unguided manner through many possible partial solutions, until happening upon a correct one. During this search, however, information about productive paths can be gleaned, and reused to make future problem solving more efficient. This speedup-learning focus is shared by Larkin's (1981) study of learning in formal
domains, in which subjects know or be able to access (e.g., in textbooks) the requisite principles for problem solving. Of course, assumptions of complete and consistent background knowledge limits the scope of predictions that can be made about human problem solving, but we will argue later that the simplification nonetheless suggests a fruitful path for psychological study, and we believe that it is amenable to perturbations that relax the assumptions of completeness and consistency.

3.2.2 Improving the Efficiency of Problem Solving

'Speedup' learning improves problem-solving efficiency by reusing past experiences. The simplest form of reuse is to use a previous solution as is. Unfortunately, a particular solution trace like the one of Figure 2(a) is tailored to only one problem – in this case, an overtake problems with two trains of 72 and 120 km/hr respectively, leaving three hours apart. However, we can generalize this solution trace by replacing constants (e.g., 72 km/hr) by variables, so that it matches problems other than the one for which it was first derived. This process of first deriving a solution to a particular problem by searching background knowledge, then generalizing it by appropriately replacing constants by variables is known as explanation-based generalization or EBG (Mitchell, Keller, & Kedar-Cabelli, 1986), since a solution trace or proof is regarded as an ‘explanation’ of the answer. A generalized solution or schema for overtake problems obtained by this process is shown in Figure 2(b). In the future, if the problem solver notices that a problem involves ‘two vehicles’ moving in the ‘same direction’ at rates $R_1 = X$ and $R_2 = Y$, $Y > X$, then the overtake schema can be reused to solve it without searching background knowledge from scratch.

Unfortunately, even after variabiliation, the applicability of this schema will be highly limited. Consider a second opposite-direction problem:

"Two trains leave the same station at the same time. One train travels 64 km/h to the south and the other travels north at 104 km/h. In how many hours will they be 1008 km apart?"

This has a solution structure almost identical in form to the schema of Figure 2(b); it differs only in the structure of the boxed subtree; rates must be added, not subtracted, to obtain a relative rate in this latter case. Nonetheless, the earlier solution cannot be used as is to help solve the new problem. In addition, the overtake schema may be more than simply useless – it may actually slow future problem solving. This happens because the solution to the new problem and the schema for the old problem share much in common. Most problem solvers will require some time examining the old schema before abandoning it as a viable option. This adds to the time required to solve the new problem. More generally, if many schemas exist and differ in small ways, then considerable time can be spent searching unsuccessfully for the correct schema, if it exists at all.

In response, Flann and Dietterich (1989) suggest the utility of generalizing over several schemas of the type in Figure 2(b). Figure 3 illustrates the general process; schemas are superimposed, and subtree structure that is not shared by all the schemas is severed. Letters represent propositions, which must match across solutions as well.
Figure 3: Acquiring generalized schemas by superimposing solutions.

For example, generalizing the schemas for \textit{opposite}\text{-direction} and \textit{overtake} problems yields a tree structure with the boxed rightmost subtree of Figure 2(b) severed.

The superimposition process results in schemas that can be reused in a wider variety of situations. Notice that a generalized schema no longer provides the complete solution structure since some of this has been 'severed', but it ideally provides a sizable chunk that can be completed by doing a small search of background knowledge. In this case a problem involving two vehicles with arbitrary rates would signify the relevance of the generalized schema; background knowledge would only be used to determine how the relative rate should be determined based on whether actual directions of the vehicles implied that they were moving in the same or opposite directions.

It is apparent that something like this schema generalization process is required for effective reuse, but care must be taken in how the procedure is applied. For example, consider a \textit{round-trip} problem in which one vehicle goes from point to point at one rate, and returns at another. In this case, relative rates are not applicable to computing trip time. Applying the superimposition method to a schema for \textit{round-trip} problems and a schema for either \textit{overtake} or \textit{opposite direction} or both, would result in a trivial schema: $T = D/R$. Little or no problem-solving advantage is gained by its derivation.

Our examples illustrate some important principles: overspecialized schemas can actually detract from problem solving efficiency, since many distinct but similar schemas introduce redundancy into the search for past experiences that are relevant to new situations. In contrast, overgeneralization results in schemas that are underconstrained and provide little or no benefit. These principles correspond exactly to those of intra- and inter- category overlap. Neither overspecialization or underspecialization is desirable; in problem solving, as in categorization, there appears to be an ideal level of abstraction for schemas and concepts. Roughly, these same principles were enumerated earlier by Gick and Holyoak (1983; pp. 8–9). In particular, they argued and experimentally supported the idea that analogical transfer to a new problem is best facilitated by a schema that captures the 'optimal' commonality between two previous source problems. Reed (1989) stressed a corollary that the source problems may be such that any generalization will be an overgeneralization, thus providing no benefit; Reed posits that this was the case in his studies, thus explaining a lack of transfer due
Figure 4: An abstraction hierarchy over problem schemas. Adapted from Yoo and Fisher (1991a, b).

to schema abstraction on selected algebra story problems.

One methodological strategy for discovering abstract schemas for problem solving is to define an analog measure to category utility for problem solving. In fact, we will turn to this task in Section 4.3, but initially we will describe a method that stores schemas at many levels of abstraction, and with experience discovers those that are most useful. In this way the system gradually converges on schemas that are optimal relative to the requirements of the environment.

### 3.3 EXOR: Improving Problem Solving by Concept Learning

Yoo and Fisher (1991b) defined a system called EXOR (EXplanation ORganizer), which forms an abstraction hierarchy over a stream of problems and their solutions. For instance, in the domain of algebra story problems an abstraction hierarchy over 48 problems drawn from 12 problem types enumerated by Mayer (1981) is formed like the one shown in Figure 4. Associated with each node is a generalized schema like those described previously. The schema at each node is shared by all the schemas stored below it; put another way, each schema extends its ‘parent’ schema in a unique way.

#### 3.3.1 An Example of Problem Solving by Categorization

The advantage of this abstraction hierarchy is that it constrains problem solving search. Figure 5 illustrates how this process operates on a specific example. In step (1), a problem statement is presented along with a quantity that must be computed. The problem statement is compared against the schemas at the first level of the abstraction hierarchy. In this case, we want to solve for ‘time’ so a first level node that solves this quantity is selected (as opposed to ‘distance’ or ‘rate’). The general ‘schema’ that ‘time’ can be solved by dividing ‘distance’ by ‘rate’ is asserted as relevant to the current problem, and this becomes the current hypothesis. In step (2) the schema is
specialized further by categorizing the problem relative to one of the children of the current hypothesis. We will describe how a child is selected shortly, but suffice it to say that one is chosen, and its schema extends the current hypothesis. This extension to the current hypothesis becomes the new hypothesis. The specialization process continues in steps (3) and (4), but in step (4), a contradiction between the surface features of the new problem and the surface features present in the highly-specific schema is found. In particular, the new problem involves one car going ‘east’ and the other going ‘west’, whereas the specialized schema requires two cars going ‘west’.

Thus, the new problem is incompatible with the hypothesis. This hypothesis is retracted, and one of its siblings is chosen as the current hypothesis in step (5). However, no extension of schema $B$ will be applicable for this reason, though our problem solver has no way of knowing this in advance of investigating all of $B$’s specialized schemas. After all extensions of $B$ are exhausted, a final attempt to complete $B$’s schema using background knowledge is made, but this will not work for the reasons stated. Thus, attention returns to $B$’s parent and a sibling of $B$ is chosen as the next hypothesis in step (6). This hypothesis represents the class of problems with vehicles moving in opposite directions; one of its extensions, represented as an existing schema or found by a search through background knowledge if no appropriate extension has been previously encountered, will solve the current problem.

3.3.2 Categorization and Learning

Figure 6(a) generalizes our example; it illustrates that a new problem will be classified down an ‘appropriate’ path of the hierarchy. At each node along the categorization path, the schema associated with the node is asserted as participating in the solution of the problem being classified. If at any point, a condition known from the problem statement contradicts one of the conditions asserted at a node, then the partial schema of the node is retracted, and control returns to the node’s parent where another child is investigated. If all children of a node fail to complete the node’s partial schema then an attempt is made to complete it by reverting to background knowledge (i.e., the dashed, triangular nodes of Figure 6). If this fails, then the node fails and control is returned to its parent as described before. The general categorization procedure is quite simple; as in COBWEB, inferences are accumulated as one descends the hierarchy, but unlike COBWEB each inference is a complex schema extension that can be retracted and redirected if necessary.

In describing the procedure above, we omitted two important points: how is categorization guided, and how are new schemas added to the categorization tree? To improve efficiency over a simple search of background knowledge, categorization must be well directed. An initial strategy guides a problem’s classification using the conditions that are given in the problem statement. These correspond to surface features – e.g., the fact that a problem involves ‘two’ (versus ‘one’) ‘cars’ (versus ‘boats’) going ‘east’ and ‘west’ (versus ‘north’ and ‘south’). Section 3.1 pointed out that surface similarity is often correlated with problem types and solution strategies. In these cases, some gain in efficiency can be realized. In particular, EXOR computes category utility over the surface conditions, which provides a degree of match between a new problem and each node of the categorization hierarchy. The nodes of a level are ordered by this match
Find: \[ T = \frac{D}{R} \]

Problem statement facts:
- east(car1), west(car2), R1=55, . . .

Figure 5: An example of problem solving by categorization.
score and investigated in order until a complete solution is found. Notice that to compute this score we need to know the distribution of surface features among problems that are stored in the hierarchy. Thus, probability distributions over a subpopulation’s surface features are stored at a node, in addition to background structure that is true of all of a node’s descendents.

If a problem is solved by appealing to background knowledge at a particular node, then the solution is added as a new child (i.e., schema) to the tree at this point. If an existing schema solves the problem, then no change to the categorization structure is made, other than to update the probability distribution of surface features at nodes under which the problem was finally placed. In some cases the new solution is generalized with existing schemas as well, thus creating new schemas. In the tree of Figure 6(a), the solution to a new problem would be added as indicated in Figure 6(b). Thus, the system incrementally clusters problem-solving solutions, and creates generalized schemas in the process.

### 3.3.3 Empirical Results

To test the merits of this problem-solving strategy, 32 training problems were incrementally added to the categorization hierarchy. At intermediate points in training, a separate set of 16 test problems were solved via categorization with the hierarchy. The average cost of solving these test problems was measured by the number of features that were instantiated (matched) during categorization. Roughly, this is the sum of the surface and background features in all the schemas successfully and unsuccessfully hypothesized in the course of solving a problem. This measure of cost is well-correlated with the time required to solve the problem, but it does not depend so strongly on the efficiency characteristics of a particular computer implementation.

Figure 7 shows the learning curve averaged over 10 random orderings of the 32 training problems. The declining upper curve illustrates a decrease in the total cost of problem solving; the darker, increasing curve represents the subset of instantiations performed just in matching features that are present in nodes of the hierarchy. The difference in the two curves is the number of instantiations performed using primitive background rules in an attempt to complete generalized schemas. Thus, total problem-solving effort declines with training, but an increasing amount of that remaining effort is placed on the growing categorization hierarchy.
Figure 7: Performance as a function of training. Adapted from Yoo and Fisher (1991a, b).

In sum, EXOR builds on the basic COBWEB strategy of top-down categorization and learning. Each node corresponds to a chunk or schema, which is a partial solution common to all category members. These problem solving 'chunks' are not used to guide categorization, but upon making a categorization, are asserted as applicable to a new problem. Rather, categorization relies on probabilistic distributions over the surface features.

3.3.4 Inferred Features and Pruning

The gains in EXOR's problem-solving performance can be improved in two ways. As we noted in Section 3.2.2, overly-specific schemas can detract from problem-solving efficiency. An example of this is illustrated by schema C of Figure 5. In this case it is not important that two vehicles are both moving 'west' (or both move 'east', 'north', or 'south'). It is only important that they are moving in the same direction. Likewise, for opposite-direction problems it is only important that two vehicles are moving accordingly; the particular directions do not matter. Such schemas can differ in very small, but nonetheless significant ways from novel problems, thus 'misleading' categorization for a large population of new problems. Therefore, each node of the categorization tree maintains two counts: a count of the number of times that the node was visited during problem solving, and the number of problems successfully solved under the node. If the ratio of successful visits to total visits drops below a threshold (e.g., 0.5), then the node and its descendents are 'pruned' from the categorization tree. Figure 8(b) illustrates the effect of this pruning. If all of a node's children are pruned, then an attempt to complete the node's schema appeals directly to background knowledge.

EXOR's problem-solving efficiency can be further improved by focusing on inferred
features. EXOR categorizes problems based on facts presented in a problem statement, but recall that Chi et al. (1981) found that ‘experts’ categorize problems based on principles that play an important role in a problem’s solution (e.g., problems with solutions that depend on Newton’s third law are categorized together).

As with surface features, useful inferred features are those that are discriminating of paths that should be followed during categorization. In general, these will be features, $F_k$, with high cue validity $P(C_i|F_k)$, relative to a schema category, $C_i$. Recall that cue validity is an important component of category utility. The more predictive a feature is of a particular path or small number of paths, the more efficiently search of the categorization tree will be directed. For example, as we noted above, surface features indicating that one car is going ‘east’ and one car is going ‘west’ are of little predictive value, but an abstract feature that is highly predictive of a solution strategy is the fact that the two cars are moving in ‘opposite direction’s. However, inference of a feature comes with a cost. Thus, we wish to weigh the predictiveness of a feature and cost savings in categorization due to its inference, against the cost that will be required to infer it.

We formalize this tradeoff in terms of the expected number of problem-solving steps (or features instantiated, or any of several other measures of cost) required to solve a problem with and without knowledge of a feature’s truth. Let $EC(C_i)$ be the expected cost of solving an arbitrary problem beginning at node $C_i$ of an EXOR categorization tree, $EC(C_i|F_k)$ is the expected cost required to solve the problem if a (derived or surface) feature $F_k$ is known, and $EC($prove $F_k)$ is the expected cost of proving (deriving) the feature. In the case of surface features, this latter cost is zero. Putting these quantities together, it is useful to verify a feature at a particular node $C_i$ if

$$EC(C_i) > P(F_k|C_i)[EC(C_i|F_k) + EC($prove $F_k)]$$
$$+ [1 - P(F_k|C_i)][EC(C_i|\neg F_k) + EC($prove $\neg F_k)],$$

where $P(F_k|C_i)$ is the probability that we will be able to prove the truth of $F_k$ (e.g., ‘opposite-direction’) and $1 - P(F_k|C_i)$ is the probability that its complement (e.g., ‘same-direction’) is true, which can also be predictive of a particular course of action. Intuitively, the more ways there are to infer a feature in background knowledge, the more possibilities that must be examined, and the greater the cost of inferring it.
EXOR identifies features that are good candidates for inference at each node in the categorization tree. Having made such identifications, statistics on these *cost effective* background features are maintained and exploited in the category utility calculation that was originally limited to surface features. Yoo and Fisher (1991b) report that the additional guidance provided by these inferred features, and the pruning method described above, collectively improves problem-solving efficiency by approximately 12% over the results of Figure 7.

### 3.4 Summary

COBWEB's approach to categorization and learning has shaped our view of problem solving, but with some important caveats. Notably, inferences are still facilitated by categorization, but schemas composed of both surface and background features are inferred and accumulated into a complete solution as a problem is categorized relative to past experiences. EXOR's categorization appears consistent with the findings and ideas of Chi et al. (1991), Ross and Spalding (1991), and others: the system infers background features that guide categorization in problem solving, but surface features are still used if they are predictive.

Furthermore, the evolutionary process underlying EXOR may also account for other data on the transition from novice to expert problem solving. Larkin (1981) found that novices tended to reason 'backward' from the goal, but that experts reasoned forward from the known facts. This forward-reasoning characteristic among experts was also found by Koedinger and Anderson (1990) in geometry problem solving. However, expertise in other domains such as design and troubleshooting (or diagnosis) suggests a dominant backward-reasoning component (Perez, 1991). As Larkin suggests, EXOR begins as a backward reasoner: hypotheses are extended from the goal through categorization and/or background knowledge search until a solution consistent with the known facts is found. However, as cost-effective background features are identified they are inferred at appropriate points in categorization by forward reasoning from the givens. We speculate that forward reasoning among experts is dominant in some domains because the top-level goals themselves are deemed cost-effective background features. This may help explain the variance of the forward versus background phenomena across domains, and suggests that in many domains a combination of the two is most appropriate.

As a cognitive model EXOR is quite tentative and we have noted that it is oversimplified in that it assumes complete and correct background knowledge, but it nonetheless suggests a line of research. While psychologists have uncovered plentiful evidence on the importance of inferred features in categorizing problem-solving experiences, we know of no work that identifies the criteria that humans use to select these features. Rather, a starting point in the search for these criteria is suggested on computational grounds: if humans are bounded-rational agents then they may select inferred features that are *cost-effective* — relatively inexpensive to infer and effective at discriminating relevant past experience.
4 General Discussion

This chapter argues that categorization guides inference in problem solving, though the role, dominance, and form of categorization undoubtedly varies with domain (e.g., van Gelder, this volume). This section takes a step back, views our models within the larger literature on categorization and problem solving, and speculates on what the unification of categorization and problem solving buys us in terms of suggesting future exploration in both areas.

4.1 Categorization and Inference

Smith and Medin (1981) point out the dual roles of categories and concepts: they allow data compression through categorization, and subsequent inferences about unobserved aspects of the individual. In COBWEB, probabilistic distributions over surface features are the sole basis of both categorization and inference. The tradeoff between categorization and inference is represented in category utility by cue validity, $P(C|V)$, which promotes identification of categories from features, and category validity, $P(V|C)$, which promotes inference of features from categories. However, COBWEB is a degenerate case of more sophisticated categorization and inference schemes that distinguish both surface and background knowledge.

In Smith and Medin’s analysis, a category’s identification procedure (i.e., categorization) relies exclusively on surface features (or perceptual features in their terms), and a category’s core contains those surface or abstract (i.e., background) features that are inferred after identification. In addition, the findings of Chi et al. (1981) and others indicate that some abstract features are inferred prior to and then guide categorization. Thus, identification too may well involve both surface and abstract features.

The dualism between identification and core is manifest in the representation and processing of schemas in EXOR. A probabilistic representation of surface features and cost-effective background features supports identification, but a logical conjunction of necessary and jointly sufficient features represents the core – those features that are inferred. Both psychological evidence and computational advantage speak for this dichotomy. Several studies on goal-derived (Barsalou, 1985) and ‘function-motivated’ (Ahn, 1990) categories indicate that categories are formed around a common function, action, goal, etc., but that principles of family resemblance often capture the surface structure of such categories. In fact, it is often not possible to conjunctively represent the various ways of satisfying a common goal; probabilistic representations of surface features allow flexibility representationally, but provide more constraints than a logical disjunctive representation, thus better supporting discrimination between contrast categories (Fisher & Pazzani, 1991).

The debate between necessary and sufficient (or classical) and probabilistic representations is a long one (Smith & Medin, 1981). The most interesting aspect of current debate (Averill, this volume; Malt, this volume) is that the lines drawn between classical and alternative representations (e.g., probabilistic) seem closely, though not perfectly aligned to the roles of surface and background knowledge in identification and inference. Some misalignment is implied by EXOR’s probabilistic assessment of certain cost-effective background features, and other models allow probabilistic rep-
resentations of background, as well as surface features (Pazzani, 1990). Thus, it is likely that the classical/core and probabilistic/surface dichotomy is oversimplified, but a natural generalization may be realistic: features increase in certainty, often but not necessarily to the point of logical certainty, as one moves from the surface to highly goal-oriented, background features.

### 4.2 Learning Issues

There are many aspects of learning to solve problems effectively. One aspect that we have not examined is improving the accuracy of problem solving in the face of inconsistent and incomplete background knowledge; EXOR assumes that background knowledge is sufficient to generate complete and correct solutions. The utility of schemas constructed in this way is then empirically tested as new problems are solved, and low utility schemas are pruned. In a suitable extension to EXOR, this generate-and-test strategy can undoubtedly be adapted to prune incorrect schemas as well, provided that an external ‘teacher’ identifies incorrect solutions. In addition, several researchers are addressing issues of incompleteness. For example, Mooney (this volume) uses a generate-and-test strategy like EXOR’s to evaluate the merits of knowledge derived from background, but patterns discovered in observations are also used to fill gaps in background knowledge, thus extending the capabilities of the problem solver. This general approach also appears in work on student modeling for intelligent tutoring (Sleeman, Hirsh, Ellery, & Kim, 1990): new (correct and ‘buggy’) rules arise when a system (or human student) bridges a procedural gap in problem-solving knowledge. Similarly, research in knowledge acquisition (Bareiss & Slater, this volume) looks to human experts to fill the reasoning gaps of the automated system.

Work on EXOR has concentrated on two facets of speedup learning. One aspect is ‘chunk’ acquisition; each schema is a composite of several inference rules, and as such it takes a relatively large step towards a solution. This aspect of speedup learning is related to considerable work on learning macro-operators (Iba, 1989) and knowledge compilation (Anderson, 1983; Laird, Rosenbloom, & Newell, 1986). However, to be helpful these chunks must be applied in appropriate circumstances. This second aspect of speedup learning improves the identification procedure (Langley, 1985). In particular, EXOR’s probabilistic representation and category utility-based categorization procedure search out reliable associations of surface and cost-effective features with the core features of the schema.

If reliable associations are not found, because the schema is retracted an inordinate number of times in the course of solving new problems, then the schema is pruned. This ‘forgetting’ phenomenon relies on a retrospective analysis of a schema’s utility, but psychological studies indicate that some categories may be prospectively pruned as well. Consider ad hoc concepts (Barsalou, 1983), which are goal-derived categories, but which are nonetheless transient, in part because they lack an efficient identification procedure. For example, things to remove from the house during a fire might include ‘family pictures’, ‘silverware’, ‘coin collections’, and ‘important documents’. These items are related by very abstract features such as ‘high value’, ‘irreplaceable’, and ‘easy to remove’, but they lack similarity ‘close to’ the surface level on which identification relies. EXOR does not model this prospective version of pruning, but undoubtedly
forgetting in this case is desirable.

Unfortunately, the lack of a good identification procedure is often an unintentional consequence of nonoptimal training strategies. Bransford, Sherwood, Hasselbring, Kinzer, & Williams (1990) describe the ubiquity of inert knowledge, which a subject possesses, but which is not spontaneously accessed in relevant contexts. For example, a subject may know how to manipulate logarithms (i.e., core knowledge), but they may not apply this knowledge when presented with large multiplication problems that would be considerably simplified by logarithms, apparently because they lack a suitable identification procedure. Thus, Bransford et al. (1990) advocate ‘anchoring’ the instruction of problem solving knowledge to well-defined cases. This type of anchored or case-based instruction is central to EXOR’s learning capabilities. Cases or problems are presented, solved, stored, and generalized as learning proceeds. While the cases that are used are impoverished compared to those advocated by Bransford et al. (1990), the requisite condition on them for effective learning remains the same: these problems must contain sufficient context so that the learner acquires an effective identification procedure.

The importance of improving the identification procedure is illustrated further by the problem-solving fan effect, which was identified by Shrager, Hogg, and Huberman (1988). They developed a formal, graph-theoretic model that learned chunks of problem solving knowledge much like EXOR. Their basic finding was that if the identification procedure does not improve as the number of learned chunks increases, then overall performance degrades. In effect, learning increases the contrast set of schemas and the overall inter-category similarity between schemas, thus confusing a less than perfect identification procedure. This phenomenon is a different and refined view of the fan effect described by Anderson (1974): if the identification procedure does not ‘improve’ in the presence of increasingly similar stimuli, then the most idiosyncratic stimuli will be recognized more quickly. It is not difficult to show that this behavior is nonoptimal. If our goal is to minimize problem-solving effort over a population of problems, then it is desirable to more efficiently handle problems with more common solution patterns.

4.3 Abstraction and Basic Levels of Problem Solving

One way to mitigate fan effects is to explicitly modify the identification procedure as chunks are acquired, but an equivalent way reorganizes the knowledge on which the identification procedure is applied. By forming schemas from common portions of individual solutions, we remove redundancies in the search for relevant problem-solving experiences. Given a choice between contrast schemas, rational approaches will favor those that cover a larger proportion of the environment, as represented in category utility by the \( P(C_k) \) term. That is, common patterns will be favored over idiosyncratic patterns, unless the idiosyncratic schemas allow for considerably larger steps to be taken towards a final solution. As we noted in Section 3.2.2, problem solving aspects of redundancy and step size correspond exactly to principles of intra- and inter-category overlap. Just as there is a basic level that optimizes a tradeoff between these concerns in object memory, there is an optimal level of abstraction in problem solving.

Abstractions have long been recognized as important in guiding problem solving (Korf, 1987), but there is little research on assessing ideal or optimal levels of abstrac-
tion in this context. Exceptions include Morris and Murphy (1990) and Rifkin (1985), each of whom found that subjects identify one level as basic in event hierarchies for reasons very similar to the principles that predict basic levels in object memory. Our own research suggests that basic levels enable an agent to minimize problem-solving cost (e.g., time, number of steps). For example, in communicating directions to a lost driver, a conscientious observer will attempt to minimize the expected effort of the driver by providing 'well'-spaced landmarks between the current location and final destination. Too few landmarks increase the likelihood of a 'wrong turn', while too many may tax the driver’s short-term memory, thus inviting mistakes as well.

This spatial example corresponds nicely to a graph-theoretic model of Shrager et al. (1988), where a schema is a set of selected states along one or more paths from an initial state to a goal state. A schema that lists all states along one path (i.e., a solution) directs problem solving perfectly, but it is not applicable to other situations. Inversely, too few states may share many possible paths, and thus be widely applicable, but relatively unhelpful in constraining problem solving. We define a basic level schema to optimize a suitable tradeoff between applicability and constraint, and propose category utility as a tentative starting point in the search for a basic level predictor in problem solving:

\[ P(\text{schema}) \sum_j \{P(S_j|\text{schema})^2 - P(S_j)^2\}, \]

where \(S_j\) are states or subgoals that are indicated by the schema. This balances the applicability of a schema, which can be computed as the proportion of paths that pass through the schema's designated states, with the expected number of states that will be correctly predicted to lie on the final solution path. In the lost-driver example, this latter aspect corresponds to our ability to predict whether landmarks were passed or not by the driver, in addition to those supplied by the directions.

It may seem counterintuitive to adopt a measure of expected accuracy like category utility to predict basic levels of problem solving, which we define to minimize expected cost. In fact, an analog can be formalized that measures the expected cost of finding a correct solution (i.e., path). Intuitively, the categories (identified by category utility) that facilitate more accurate predictions with constant cost (i.e., one prediction per state), correspond closely to categories that insure a correct final solution (i.e., constant accuracy) with less expected cost.

In apparent contrast to our advocacy of basic levels, case-based reasoning models (Kolodner, 1987) superficially focus on how previous cases (i.e., instances, solutions) are best accessed and exploited in problem solving. Case-based reasoning should not be confused with case-based instruction, since the vast majority of models in machine learning presume that learning advances by analyzing specific cases, though the role of cases is to form abstract schemas that are then used for problem solving.

Moreover, despite the label, the vast majority of case-based systems form and exploit abstractions for categorization (e.g., through indexing) and inference. To do otherwise might invite fan effects. Thus, the distinction between case-based versus abstraction-based reasoning can distract analysis from more general insights into optimal levels of abstraction, cases or otherwise, as well as evaluation along other dimensions such as a system's ability to cope with incomplete background knowledge or an incomplete case library.
4.4 Prescriptions for Learning and Training

From a speedup perspective, learning displaces fan effects by introducing abstractions. This displacement is illustrated in EXOR's learning curve of Figure 7; initially, chunking diminishes performance until appropriate abstractions and a reliable identification procedure are formed. Ideally, a learner will evolve towards a basic level of description, which may, like EXOR and COBWEB, be further decomposed into more specialized concepts and schemas that are helpful in increasingly restricted contexts.

In general, we believe that our study of basic levels suggests a more fundamental analysis of 'similarity' between problem schemas. Such an analysis may shed light on a number of phenomena. For example, it is well-known that there is little or no transfer between isolated problem-solving experiences and similar, if not identical, new problems by novice subjects (Reed, Dempster, & Ettinger, 1985). EXOR predicts this, since an overly-specific schema may be examined, but will not be transferred if it differs in the smallest way from a new problem. In fact, no meaningful transfer occurs until a 'contrast' problem solution is examined and is generalized by superimposition with a previously cached solution. More generally, Gick and Holyoak (1983) argue that analogical transfer is greatly facilitated by a schema that is derived from at least two source analogs, since it abstracts out unimportant detail in each source, thus making the 'similarity' between a new problem and the helpful aspects of the previous problems more apparent.

We can also use principles of similarity to prescribe training schedules. For example, EXOR and COBWEB (Fisher, et al., 1992) master a domain most rapidly when problems of high 'contrast' are presented repeatedly in sequence; learning is slowest when similar problems are presented back to back (i.e., as one might do when practicing a particular problem type before moving on to another type), though our computer model does not take into account short-term memory limits that might alter this preference. In addition to between-problem considerations, within-problem variations can promote effective learning as well. Notably, our lost-driver analogy argued that knowledge could be best communicated and reused by an appropriate spacing of landmarks, states, or subgoals (Ruby & Kibler, 1991). Simply presenting a student with a problem may cause a novice to flounder, thus requiring considerable effort to solve each problem. In contrast, presenting a student with complete solutions will not facilitate generalization to other problems, thus requiring more problems to master a domain. Thus, Julio Ortega (personal communication) suggests that presenting subjects with basic level schemas, which must then be completed by the student, will tend to minimize the overall effort required to master the domain, where effort is the product of the number of problems and the effort per problem required before a subject reaches a certain level of mastery.

In sum, the gradients implied by basic levels and similarity issues more generally suggest quantitative measures that can be used to assess, predict, and prescribe training schedules.

5 Concluding Remarks

This chapter argues the merits of a unified view of categorization, concept learning, and problem solving. This view is advantageous to the extent that it suggests promising
lines of research. Notably, we have argued that the transition from novice to expert problem solving is mediated by principles of effective categorization, and that the early dominance of problem-solving fan effects is gradually displaced as one converges to basic levels of problem solving. Metrics developed to assess category and schema quality can be used descriptively to track the course of learning, or prescriptively to direct it.

In closing, it is worth highlighting several methodological biases that stem from our view of cognitive modeling as a design task. Most prescriptions of design assume an initial specification of behavior, perhaps from several experimental studies, and the formulation of objective functions that specify desirable aspects of the input and output of the final product. Anderson's (1990) ideas of a rational analysis represent such a step, and in fact, Anderson traces these ideas to Marr (1982), who was influenced by design issues in the information-processing paradigm. The vital point is not that rational analyses are new per se to cognitive modeling, since others (e.g., Gluck & Corrter, 1985) have adhered to these principles before, or that Anderson's specific (i.e., Bayesian) formulations are 'correct', but that Anderson expounds a methodology that is rarely made explicit, even by those that use it: a reasonable starting point for cognitive modeling is a procedure that optimizes a suitable tradeoff between cost, correctness, and other aspects of bounded rationality.

The rational procedure offers several advantages, two of which we describe. The first recognizes that our ultimate goal is a process model, which commits to particular mechanisms. However, deficits of the process model can often be better understood and corrected by appealing to the higher level specification. For example, Richman (1991) describes a process model based on EPAM, which accounts for the same typicality data as presented in Section 2.2.2, but category utility suggests certain refinements to EPAM's attentional mechanisms, which might yield a better fit to the data. Second, for those at the interface of cognitive psychology and artificial intelligence, a rational procedure can be exploited in artificial (e.g., engineering) environments that enforce the same tradeoffs as the natural system that motivated the analysis. For example, COBWEB has been adapted as a 'clustering' and data analysis tool (Mitra, this volume) for engineering applications (Fisher, Xu, Carnes, Reich, Fenves, Chen, Shiavi, Biswas, & Weinberg, 1991), as well as for cognitive modeling.

Finally, cognitive modeling is novel relative to other design applications in at least one very important respect. In many design applications, moving beyond the known data or specification is undesirable. In contrast, cognitive models are maximally helpful when they move beyond the data, thus pointing the way for further exploration. We have touched upon several ways in which COBWEB and EXOR do this, though in each case, but particularly with respect to EXOR, we have only tentatively embarked on the iterative process of refinement necessary for a robust understanding of categorization and problem solving.
Acknowledgements

We thank the editors, Glenn Nakamura, Roman Taraban, and Doug Medin, for thorough and influential comments on earlier drafts. In addition, Laura Novick provided helpful comments and suggestions on correctness and style that went beyond the call of duty, and which we have tried to accommodate. Section 2’s discussion of COBWEB is detailed further in Fisher (1987) and Fisher and Langley (1990). Discussion of EXOR, notably Section 3.3, appears in expanded form in Yoo and Fisher (1991a, b). This research was supported by NASA Ames grant NCC 2-645.
Endnotes

Endnote 1 (p. 4): A feature's majority value at a node is used for prediction. This probability-maximizing strategy contrasts with the probability-matching assumption underlying \( CU \). Nonetheless, \( CU \) favors categories that are better suited for prediction, even when predictions are actually generated by a probability-maximizing strategy (Fisher, 1987a).
References


INDEX TERMS

Abstraction
Ad hoc concept
Algebra story problem
Analogical reasoning (or Analogical transfer)
Anchored instruction
Artificial intelligence
Background knowledge (or Domain theory)
Backward reasoning
Basic level
Basic level effect
Basic level of problem solving
Bounded rationality (or bounded-rational)
Case-based reasoning (or case-based learning)
Case-based instruction
Category
Category utility
Category validity
Categorization
Categorization hierarchy
Chunk
Classical concept (or representation)
Clustering
COBWEB
Cognitive model
Concept formation
Concept learning
Concepts
Core (of a concept)
Cost-effective feature
Cost-effective inference
Cue validity
Data analysis
Domain theory (or Background knowledge)
EBG
EPAM
EXOR
Expert
Explanation-based generalization
Explanation-based learning (or Explanation-based generalization)
Fan effect
Family resemblance
Forgetting
Forward reasoning
Goal-derived category (or concept)
Hierarchical categorization
Identification procedure
Induction
Inert knowledge
Inference
Intelligent tutoring
Knowledge compilation
Machine learning
Macro-operator
Negative fan effect
Novice
Perceptual feature
Probabilistic concept (or representation)
Probabilistic concept tree
Probability matching
Probability maximizing
Problem solving
Problem-solving fan effect
Process model
Production system (or Production-system model)
Rational analysis
Recognition
Schema
Search
Similarity
Sorting
Speedup learning
Student modeling
Supervised learning
Surface feature
Typicality effect
Unsupervised learning