Modeling Inter-Category Typicality within a Symbolic Search Framework

Craig S. Miller
Computer Science Department
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213-3891
cmiller@cs.cmu.edu

Abstract

This paper addresses category typicality in the context of a category naming task. In contrast to the predominant effort with gradient models, a symbolic search framework is taken. Within this framework, the SCA (Symbolic Concept Acquisition) model demonstrates varying response times as a function of an instance's intra-category typicality. Here its coverage is expanded to inter-category typicality. A functionally motivated extension for SCA is advanced that pursues search backtracking under ambiguous cases. I explain how the backtracking extension accounts for inter-category typicality effects, and support it with some empirical evidence. I discuss how the effect generalizes to a larger class of symbolic search models.

Introduction

Within the last several decades, human categories have come to be characterized as flexible structures that generally lack rigid boundaries. This flexibility was revealed by experiments showing increased processing efficiency for certain category members. For example, humans can categorize "typical" category members faster and more accurately than less typical category members (Rosch & Mervis, 1975; Rosch et al., 1976). To account for these results, researchers have advanced many "gradient" models, employing conceptual structures that explicitly encode probabilities (e.g. Fisher (1987) and Anderson (1991)) or activation levels (e.g. Kruschke (1990) and Gluck and Bower (1988)).

This paper offers a contrasting perspective. In place of explicit membership gradients, conceptual membership is represented as a process where some category members require more computational resources than others. Using the framework of the Problem Space Computational Model (PSCM) (Newell et al., 1991), the work casts the processing of concept membership in terms of search through a problem space. The ongoing challenge is thus a characterization of a model that appropriately requires more search for some category members than others.[1]

Previous work has advanced one candidate PSCM model, called SCA (Symbolic Concept Acquisition) (Miller & Laird, 1991), motivated by the Soar architecture (Newell, 1990), a computational implementation of the PSCM. Analysis and empirical results reveal that SCA already exhibits a range of typicality effects measured in terms of processing time and accuracy (Miller, 1993). In this paper, I go beyond this work by describing a functionally and architecturally motivated extension to SCA that expands its coverage of typicality effects to *inter-category* typicality.

SCA and Typicality Effects

The Problem Space Computational Model (PSCM) treats problem solving as search through a space of states. An *operator* effects the transition from one state to another by modifying the existing state. In previous work, Miller and Laird (1991) cast the task of category prediction in terms of the PSCM framework. Each state corresponds to an object description and the problem space to the set of all possible object descriptions. Operators incrementally modify the object description until a recognizable state is produced. At this point, an operator, acquired from prior experience, names a category.

SCA (Symbolic Concept Acquisition) uses *abstraction* operators to effect the transition between states. These operators incrementally remove features from the object description until a naming operator can apply. In short, the abstraction operators serve as a controlled means of generalizing a large space of specific object descriptions to a smaller set of general naming rules.

Consider the example in Figure 1. The task is to name the category of the object described as oblong, red, smooth, and small. For this example, we will assume that the system has already acquired some naming knowledge for a small subset of the object description space. A extensive treatment of how SCA acquires rules and search control knowledge is provided elsewhere (Miller & Laird, 1991; Miller, 1993).[2]

The presented object description serves as the first state in a search for a recognizable object description. There is no category naming rule that matches the feature description S_1. Thus, the search proceeds by applying an operator. In this example, an abstraction operator applies, producing a new state (S_2) by removing the feature `small`. Again no category naming rule recognizes the state. Search continues with the application of a second abstraction operator. This time the feature `smooth` is removed from the object description.

[1] The reader should not confuse this notion of search with the concept of "inductive search" prevalent in machine learning (Michalski, 1983; Mitchell, 1982). In this paper, the search process retrieves a category name for each individual instance whereas inductive search seeks a consistent logical concept definition for a set of pre-classified instances.

[2] While some issues such as optimizing abstraction operator selection are of immense practical importance, I will omit much of their coverage since it is not needed in analyzing the results presented here.

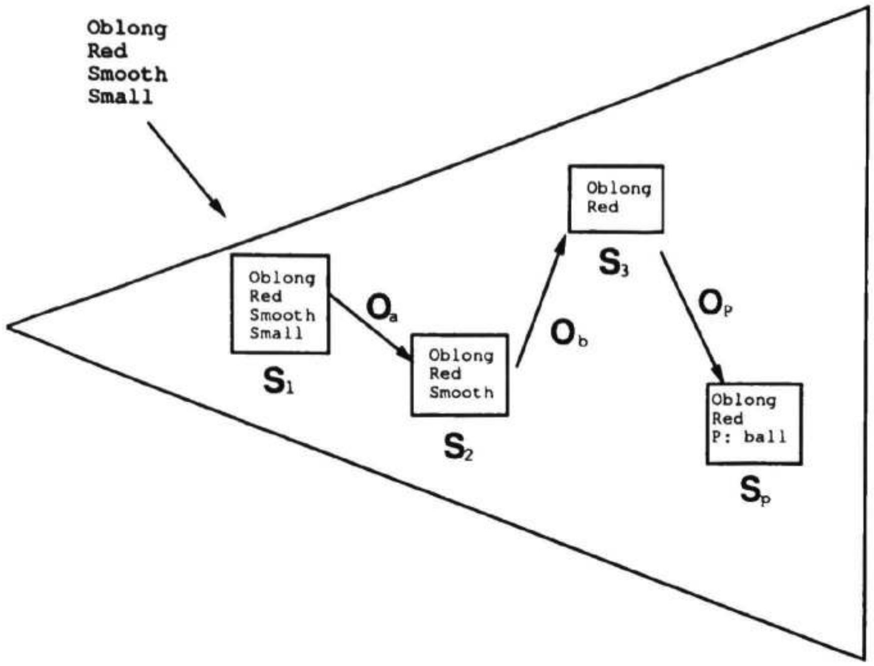

Figure 1: Category prediction cast within the PSCM

With the creation of (S_3), previously acquired knowledge recognizes the state and applies a *naming* operator that augments the state with the category `ball`.

SCA learns new, more specific rules through the application of general rules to training examples. Thus, SCA starts off with very general rules, but with experience slowly acquires more specific ones. Because SCA attempts to match the most specific rules first, a practice effect ensues with experience.

SCA's practice effect is most evident for frequent combinations of features that name a common category. Since typical members generally share features among their class, SCA realizes an intra-category typicality effect (Miller & Laird, 1991; Miller, 1993). Namely, it produces fewer errors and faster response times for the category's more typical members, a robust effect exhibited by humans (Rosch et al., 1976).

However, humans also exhibit an *inter-category* typicality effect. In contrast to intra-category typicality, defined by how similar an instance is to other members of the same category, inter-category typicality is defined by how *dissimilar* the instance is to instances of contrasting categories. People make fewer errors and faster response times for instances with higher inter-category typicality (Rosch & Mervis, 1975).

That SCA, or any other concept learning system, exhibits fewer errors with high inter-category typicality examples is no surprise, as these examples are less easily confused with examples from contrasting categories. This is especially the case for SCA, which requires sufficient experience with training examples before acquiring rules whose conditions include the necessary discriminating features.

In terms of response time, however, SCA's practice effect does not account for inter-category typicality. In explaining why, let us consider the category members, represented as small-case letters, shown in Figure 2. This figure depicts similarity as the euclidean distance between examples. For example, the close spatial proximity of examples **a** and **e** denote that they have more features in common than examples **b** and **c**.

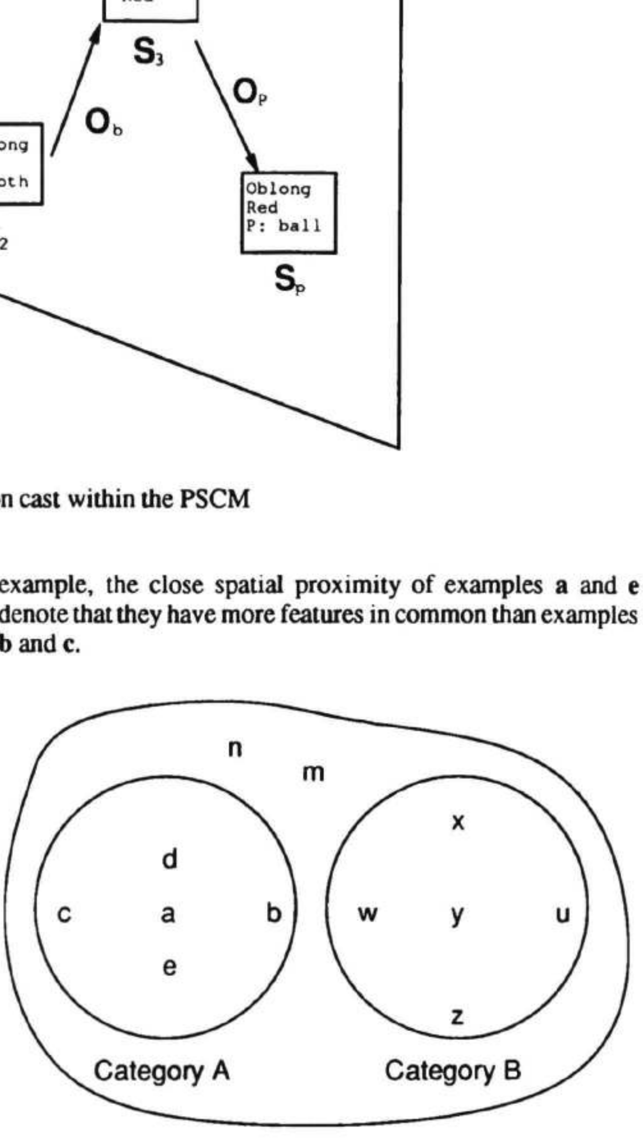

Figure 2: Spatial representation of instance similarity

In looking at **b** and **c**, we see that their intra-category typicality is the same since they are equally distant from other members of category A. However, of the two, **c** has a higher inter-category typicality because it is more dissimilar to the contrasting category members. During training, SCA receives as much practice naming category A with **b**'s features as it does with **c**'s features. With equal practice, rules with equal specificity are produced, and the time required to access them is the same. Instance **b** will more frequently access conflicting rules, in which case SCA makes a random guess, or will more frequently access rules naming category B, but in neither case does this require more time.

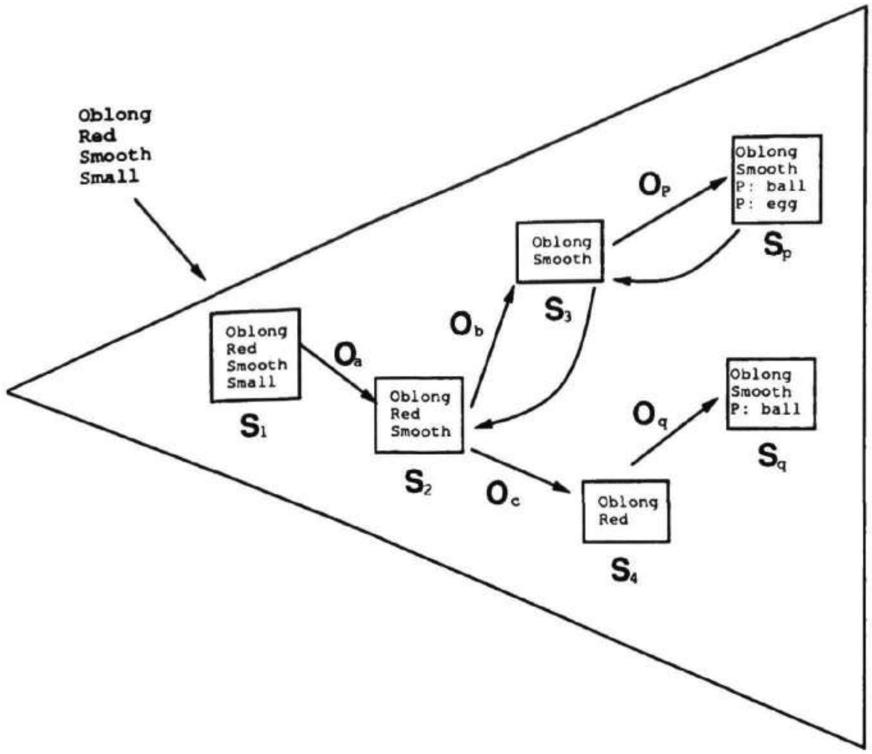

Figure 3: PSCM casting of backtracking

Extension description

Conflicts arise in SCA when search leads to a state description previously associated with two or more categories. The standard SCA implementation resolves the conflict by randomly selecting one of the categories. *Backtracking*, a recurrent technique in the PSCM and other search architectures, provides the means of making a more informed choice. Rather than randomly choosing, the ambiguity can be resolved by backtracking to a previous state, and then pursuing an alternate path in search of an unambiguous name. In the context of SCA, backtracking reverses a previous abstraction, that is, it returns to a more specific state description, before removing alternate features.

Figure 3 presents an example. As before, the model is attempting to classify the object described as `oblong, red, smooth, small`. Once again it applies an abstraction operator that removes `small` from the state description, thus producing state S_2. At S_2, it applies an operator that removes `red` from the description. Now, state S_3 is recognized, and with the application of operator O_p, *two* conflicting naming rules apply. Rather than guessing between the two category names, the backtracking implementation reverts back to S_2. From there, search follows an alternate route, removing `smooth` from the state description, which ultimately leads to one unique name.

With backtracking, the computational demands of search are sensitive to the degree of feature overlap between instances of contrasting categories. This provides a natural account of how response times vary inversely as a function of inter-category typicality. Recall that instances with low inter-category typicality share more features with instances of contrasting categories. With these overlapping features, they are more likely to access conflicting naming rules, and consequently incur a larger expense in processing time as the search backtracks through previous states.

Empirical results

Two implementations of SCA were used to obtain an empirical assessment of the impact backtracking has on inter-category typicality. The first is the standard implementation described in Miller (1993), which resolves conflicts by randomly selecting a category. The second implementation, SCA-BX, is identical to SCA, except it attempts to resolve ambiguities by pursuing one alternate path. If this likewise produces a conflict, it makes a random selection.

Both implementations use the "default" feature selection strategy described in Miller (1993). In general terms, this strategy orders feature removal based upon the selection's success in predicting one unique category name while processing training examples. During performance runs, the backtracking implementation follows this selection order for its initial search path. If this leads to an ambiguous prediction, it follows the strategy's second most prefered path. The use of backtracking potentially offers an additional knowledge source for ordering feature selection since it compares alternate selection paths for a particular instance. However, in order to avoid an additional confounding factor in comparing both implementations, this knowledge is not used here.

The two implementations were trained and tested on the data-set in Table 1. This data-set was constructed so that all compared instances have the same degree of intra-category typicality, but varied in inter-category typicality. Category A consists of the examples of different levels of inter-category

Table 1: Training and testing data for inter-category typicality.

Category	Attributes D1	D2	D3	D4	D5	Overlap Score	Typicality Group
A	c	b	a	b	b	15	Low
A	c	b	a	b	c	15	Low
A	b	a	c	b	b	9	Mid
A	b	a	c	b	c	9	Mid
A	a	a	a	a	b	3	High
A	a	a	a	a	c	3	High
B	c	b	b	b	b	–	–
B	c	b	b	b	c	–	–
B	c	c	c	c	b	–	–
B	c	c	c	c	c	–	–
B	c	b	c	c	b	–	–
B	c	b	c	c	c	–	–

Table 2: Inter-category typicality effects.

Model	Accuracy Low	Mid	High	Steps until match Low	Mid	High	Backtracking Low	Mid	High	Total steps Low	Mid	High
SCA	72%	85%	92%	2.60	2.64	2.58	–	–	–	2.60	2.64	2.58
SCA-BX	73%	87%	95%	2.63	2.63	2.56	21%	18%	13%	3.47	3.35	3.08

typicality. The overlap score is the number of features the instance shares with all of the instances in the contrast category. This score is thus the inverse of inter-category typicality. Category B serves as the contrast category.

The data in Table 1 is analogous to the data that Rosch and Mervis (1975) used for testing inter-category typicality (Experiment 6). Like the data in Table 1, their experimental examples have the same family resemblance score (intra-category similarity), that is, they equally share features belonging to other examples in the same category. Also as in Table 1, the examples were divided into three inter-category typicality groups according to the degree in which the example's features overlapped with the features of the examples in the contrasting category.

In testing each implementation, the data-set, with randomly ordered instances, was presented for five training cycles while interleaving performance trials (naming the category) after each training cycle. This process was done 5000 times.[3]

Table 2 presents the averaged performance results of SCA and SCA-BX for instances belonging to Category A. Both models are compatible with human behavior for accuracy, where accuracy is better for higher typicality. In addition, we see that SCA-BX produces a slightly higher accuracy rate for all three levels of typicality.

For SCA, response time is presented in terms of the average number of abstraction steps taken before a match occurred. These figures only differ insignificantly among the levels. The figures are essentially the same for SCA-BX. This comes as no surprise since the two implementations are identical processes through the first match.

[3]Unless otherwise noted, averaging the results over 5000 trials was more than sufficient for achieving the significance necessary for the qualitative comparisons described here.

In the next set of columns, the percentage of times backtracking occurred is presented for SCA-BX. Since backtracking only occurs for conflict resolution, instances with lower inter-category typicality were more likely to cause backtracking. The final set of columns averages in the expense of backtracking (calculated as four additional steps). Since SCA did not use backtracking, its total number of processing steps are the same as the number of steps to the first match. For SCA-BX, the additional time expense of backtracking produced faster response times for instances with high inter-category typicality as compared to those with lower typicality.

Discussion

The results of the SCA simulation with backtracking were consistent with inter-typicality effects observed in human data for the Rosch and Mervis study. Namely, instances whose features rarely overlap with instances from contrasting categories are processed faster and more accurately than instances whose features often overlap. Insight into why SCA-BX produced these results leads us to generalize the class of search models that produce these inter-typicality effects.

Since inter-category typicality is a measure of how dissimilar a category member is to those of a contrasting category, instances with a *low* level of inter-category typicality are similar to members of contrasting categories and thus possess a larger degree of ambiguity as to their proper classification. In general, *any* search model will require more processing time for these ambiguous members if the following principles hold for the model:

1. **Model has difficulty with category ambiguity.** This condition functionally necessitates further search. In the case of SCA, the model may access rules shared by instances of several categories.

2. **Model can detect ambiguity.** Before calling for further search, the ambiguity must be detected. For SCA-BX, ambiguity is detected with the retrieval of conflicting names.
3. **Model continues search in order to resolve ambiguity.** Additional time must be required in order to resolve the ambiguity. SCA-BX backtracks to a previous state, and then tries other feature combinations for retrieving a name.

Implicit with these conditions is the model's seriality. In general, it is the varying length of a sequential, deliberate search process that accounts for varying response times.

Gradient models may offer a natural approach for representing flexible category structures, as they provide an immediate account for typicality data. However, we have seen how the application of a symbolic search model provides an interesting contrast, as its account of human data emerges from a natural extension within its symbolic framework and from its functional motivation. From the perspective of a symbolic framework, the addition of backtracking to SCA is a natural and recurrent technique for search architectures. Functionally, its application improves performance. In this paper, I have empirically demonstrated how backtracking seeks out additional prediction knowledge in resolving ambiguities. Future work may also show how backtracking provides the additional functionality of learning which features to abstract first from the object description.

For this paper in particular, I have explained how the principled application of backtracking to SCA delivers response times as a function of inter-category typicality that is consistent with human behavior. In applying these data-independent principles, we thus converge on an architecturally and functionally motivated explanation of typicality.

Acknowledgements

While at the University of Michigan, this research was sponsored in part by NASA Ames contract NCC2-517. While at Carnegie Mellon University, this research was sponsored in part by the Markle Foundation. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of NASA or the Markle Foundation. I thank Roger Remington for providing valuable comments on an earlier version of this paper.

References

Anderson, J. R. (1991). The adaptive nature of human categorization. *Psychological Review, 98*, 409–429.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning, 2*, 139–172.

Gluck, M. A. & Bower, G. H. (1988). Evaluating an adaptive network model of human learning. *Journal of Memory and Language, 27*, 166–195.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review, 99*, 22–44.

Michalski, R. S. (1983). A theory and methodology of inductive learning. *Artificial Intelligence, 20*, 111–161.

Miller, C. S. (1993). *Modeling Concept Acquisition in the Context of a Unified Theory of Cognition*. PhD thesis, The University of Michigan. Also available as Technical Report CSE-TR-157-93.

Miller, C. S. & Laird, J. E. (1991). A constraint-motivated model of concept formation. In *The 13th Annual Conference of the Cognitive Science Society*, pages 827–831.

Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence, 18*, 203–226.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Newell, A., Yost, G., Laird, J. E., Rosenbloom, P. S., & Altmann, E. (1991). Formulating the problem-space computational model. In Rashid, R. F. (Ed.), *Carnegie Mellon Computer Science: A 25th Anniversary Commemorative*. New York: ACM Press: Addison-Wesley.

Rosch, E. & Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology, 7*, 573–605.

Rosch, E., Simpson, C., & Miller, R. S. (1976). Structural bases of typicality effects. *Journal of Experimental Psychology: Human Perception and Performance, 2*, 491–502.