Using Data and Theory in Multistrategy (Mis)Concept(ion) Discovery

Raymund Sison, Masayuki Numao and Masamichi Shimura
Department of Computer Science
Tokyo Institute of Technology
2-12-1 Ohokayama, Meguro
Tokyo 152, Japan

Abstract

Most conceptual clustering systems rely solely on data to form concepts without supervision; the few that exploit causalities in the background knowledge do so only after the completion of a similarity-based learning phase. In this paper, we describe a multistrategy misconception discovery system, MMD, that utilizes data and theory in a more tightly coupled way. The integration of similarity- and causality-based learning in MMD is shown to be essential for the automatic construction of accurate and meaningful misconceptions that account for errors in novice behavior.

1 Introduction

The primary method for unsupervised *concept formation* in AI is *conceptual clustering,* which is the grouping of unlabeled *objects* into *categories* for which conceptual descriptions (i.e., *concepts)* are formed. Although conceptual clusterers differ in the way they address six key dimensions,[1] they largely share the characteristic of relying solely on data to form concepts, a situation that likewise characterizes concept learning research in cognitive psychology [Komatsu, 1992]. Recent works (e.g., [Barsalou, 1991; Rips and Collins, 1993; Wisniewski and Medin, 1994]), however, reveal an increasing dissatisfaction over similarity-based models with their almost exclusive reliance on data and show an increasing interest in the role of theories and goals in concept formation.

While there may be a couple of AI systems that use data (similarity-based learning (SBL)) and theory (explanation-based learning (EBL)) to form concepts without supervision, these systems treat SBL and EBL as phases that are performed one after the other. The unsupervised learning systems of [Lebowitz, 1986] and [Pazzani, 1993], for example, first apply SBL on objects to form clusters which are then fed into an EBL or EBL-like component for further processing. The supervised learners of [Flann and Dietterich, 1989], [Mooney and Ourston, 1989] and [Yoo and Fisher, 1991], on the other hand, operate in the reverse fashion: the output of the EBL phase is sent to an SBL component. However, [Wisniewski and Medin, 1994] argue cogently that such loosely coupled approaches to using data and theory, while undoubtedly useful, remain inadequate as models of concept formation.

In this paper, we describe MMD, a multistrategy concept (or, more specifically, misconception) discovery system that utilizes data and theory in a more tightly coupled way than previous systems have. As a system that incrementally constructs and revises a hierarchy of possibly overlapping categories of relational descriptions, MMD is also unique in the manner in which it addresses the key dimensions of conceptual clustering earlier mentioned.

Misconception Discovery as a Special Form of Concept Formation Errors in novice behavior, such as bugs in a program written by a novice programmer, can be represented as logic formulas that describe specific relations, i.e., *discrepancies*, between the incorrect behavior and an ideal (Table 1 illustrates). Such sets of discrepancies are analyzed in order to uncover the underlying misconceptions that cause them. Knowledge of the causes of bugs will enable a tutor to both present a lesson and remediate a student more effectively.

In *misconception discovery,* therefore, the usual problem of concept formation is further complicated by the fact that conceptual descriptions of clusters of discrepancies can hardly be considered misconceptions — unless causal explanations for them are found.

In what follows, we first present a basic similarity-based algorithm for clustering relational descriptions and then describe how causal relationships in the background knowledge can be exploited to construct or correct descriptions of concepts/misconceptions while they are being formed. We then report experimental results showing that the approach to concept discovery embodied in

[1] These dimensions are: data processing mode (incremental or batch), data description (attributions! or relational), concept description (definitional, prototypical or probabilistic), category organization (hierarchical or flat), category overlap (overlapping or disjoint), and clustering criterion (ranging from ad hoc to Bayesian measures).

Table 1: Discrepancies in Behavior

Ideal behavior:

```
reverse([H|T],R):-
   reverse(T,T1),
   append(T1,[H],R).
```

Buggy behavior:

```
reverse([H|T],[T1|H]):-
   reverse(T,T1).
```

Discrepancies:

```
replace(head,R,[T1|H]),
remove(append_subgoal).
```

MMD enables the automatic construction of meaningful misconceptions from theory and data.

2 Incremental Clustering of Relational Descriptions

2.1 Basic Similarity Measure

Basically an object *O* is classified into a category with concept description *C* with which it has more similarities than differences. To measure this degree of similarity/dissimilarity we use as our basis Tversky's contrast model [Tversky, 1977]:

$$Sim(C,O)) = \theta f(C \cap O)) - \alpha f(C - O) - \beta f(O - C)$$

which expresses the similarity between two sets of features (in our case, discrepancies), *C* and O, as a function of the weighted measures of their common $(C \cap O)$ and distinctive *(C - 0,0 - C)* features.

We compute the commonalities between two sets of relational descriptions *C* and *O* using:

$$(C \cap O) = Com(C,O) = \bigcup_{i=1}^{m} \bigcup_{j=1}^{n} lgg(C_i, O_j)$$

where *lgg(x, y)* is the least general generalization [Plotkin, 1970; Muggleton and Feng, 1990] of atomic formulas *x* and *y* in the function-free first-order logic, and *m* and *n* are the number of atoms in *C* and O, respectively.

2.2 Basic Relational Clustering Algorithm

The basic similarity-based clustering algorithm classifies a sequence of objects into a hierarchy which it inductively revises in the process. The algorithm is *incremental,* so it takes one object at a time and classifies this object recursively into the nodes that match it to a certain degree. Each node in the hierarchy denotes a *concept,* which is either (a) a generalization (intersection or variableization) of the subconcepts below it, or (b) a record of an instance, or both. A counter is used to store the number of instances associated with a node. Table 2 describes the basic algorithm.

Table 2: Basic Clustering Algorithm

1. From the children N_1,..., N_m of a given node *N* of the concept hierarchy, determine those that *match* the set of input discrepancies, O. The *match* function computes, for every child node N_i
   - the set of commonalities, *Com(Ni,0),* between a node, TV,, and *O,* and
   - the degree of similarity, *Sim(Ni,0),* between *Ni* and *O*

   and determines whether *Sim* exceeds a system threshold, γ.
2. If no match is found, place *O* under *N.* Otherwise, for every *Ni* that matches *O,* perform one of the following depending on the value of *Com:*
   - increase the weight counter of *Ni* ;
   - replace N_i with *O* and insert *(N_i — O)* under *O;*
   - cluster *(O — N_i)* Under N_i (i.e., repeat the procedure, this time matching (O — *Ni)* with the children of *Ni);*
   - create a new node, *Com,* under TV, representing the commonalities of *O* and *Ni,* and place their differences under this node.
3. Nodes whose *(weight • height)* values fall below a system parameter may be discarded on a regular or demand basis.

The above algorithm is similar to UNIMEM [Lebowitz, 1987] and COBWEB [Fisher, 1987], which are both incremental conceptual clusterers.[2] UNIMEM's similarity measure, however, considers only the differences between two sets of features. Furthermore, UNIMEM retrieves only a set of "potentially relevant" nodes to compare against the new object (rather than examining every child of a given node), and maintains a total of 13 different parameters.

COBWEB, on the other hand, uses a probabilistic concept representation (rather than set theoretic) and a corresponding probabilistic similarity measure (category utility [Gluck and Corter, 1985; Corter and Gluck, 1992]), and can only produce disjoint clusters (but see the probabilistic clusterer in [Martin and Billman, 1994]). In terms of explaining errors in novice behavior, disjoint clusters mean that a set of discrepancies can only be classified under one "misconception", though it may well be symptomatic of several.

Both UNIMEM and COBWEB deal only with attributional descriptions, though one of COBWEB's variants, Labyrinth [Thompson and Langley, 1991], extends COBWEB to handle structured objects. Like its predecessor,

[2]The above algorithm and the *Com(C,0)* function are presented in greater detail in [Sison and Shimura, 1996a], where the algorithm is called RC. Said report also provides a more in-depth discussion of the similarities and differences among UNIMEM, COBWEB, and RC.

however, Labyrinth can only produce disjoint clusters. We also mention here CLUSTER/S [Stepp and Michalski, 1986], which is an early algorithm that handles structured descriptions by first transforming these into attribute-value form, then feeding these into an attributional clusterer (CLUSTER/2 [Michalski and Stepp, 1983]) that is nonincremental.

3 Using Causal Relations to Strengthen Coherence of Concept Descriptions

3.1 Causality in Background Knowledge

Similarity-based clusterers like the ones described or mentioned above form categories on the basis of regularities (e.g., co-occurrence, frequency) among features in the data, but ignore qualitative relationships among these same features. We argue that the presence of a qualitative, particularly causal relationship between features of a concept serves at least two purposes:

- First, causal relationships strengthen the coherence of a conceptual description, and can warrant the splitting of a concept or an object when some regularities are coincidental.
- Second, and more important for the problem of misconception discovery, causal relationships explain the regularities in the data. By examining causal relationships it is possible to gain a better understanding of the causes of raw discrepancies.

3.2 Causality heuristics

Causal relationships between features can be induced or deduced in a variety of ways. Lebowitz [1986], for example, suggests first using the frequency of occurrence of a feature in other concepts as a heuristic indicator of whether the feature is a cause or an effect, and then forward-chaining from the causative features to the other features using heuristic, low-level, causal domain rules. In [Pazzani, 1993], there are only two "kinds" of features, namely, actions and state changes, and actions are always the causative features. Determining which state changes are caused by which actions is achieved by instantiating general causal patterns.

In our case, we use causal relationships among components of the ideal behavior, together with the following heuristics:

- *Component-level causality:* Causal (or enabling or determination) relationships among the components of the ideal behavior that are present in a set of discrepancies suggest causal relationships among these discrepancies.
- *Concept-level causality:* A causal relationship between two discrepancies in a generalization node, where one is an intersection generalization and the other a variableization, suggests that the former causes the latter.
- *Subconcept-level causality:* Causal relationships between a parent node and its child suggests that the latter causes the former.

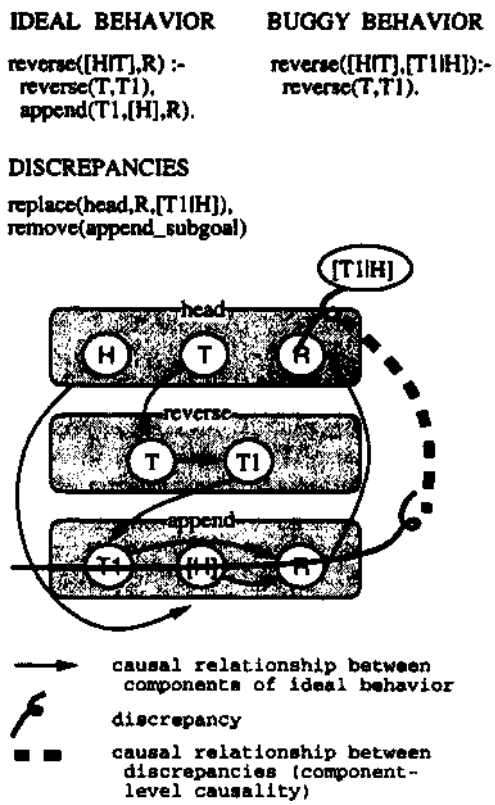

Figure 1: Causal relationships

The second and third causality heuristics assume the existence of component-level causal relationships, and are used to determine the direction of casuality.

To illustrate the first heuristic, recall the ideal program for reverse/2 in Table 1 (reproduced in Figure 1). The ideal program states that the reverse of a list is the concatenation of the reverse of its tail and its head. Note that this can be viewed as describing relationships among four objects, namely, the head *H* and the tail *T* of the list to be reversed, the reversed list R, and a temporary entity 71 representing the reverse of *T;* and the relations reverse/2 and append/3. These relationships are illustrated graphically in Figure 1.

The *component-level* causality heuristic suggests that if two discrepancies dI and (22 involve two features f_1 and /2, respectively, both of which involve a common object *c* (i.e., *c* in f_1 causes, enables or determines c in f_2, or vice versa), then *dI* and d^2 are causally related. Thus, since both discrepancies in Figure 1 involve the object *R (R* in the relation in the second discrepancy causes (enables) the R in the first), then the two discrepancies are, according to this heuristic, causally related; that is, the student's use of the construct [I] is related to the conspicuous absence of the append/3 subgoal in his program. The drawing in Figure 1 illustrates.

The *concept-level* causality heuristic suggests that if

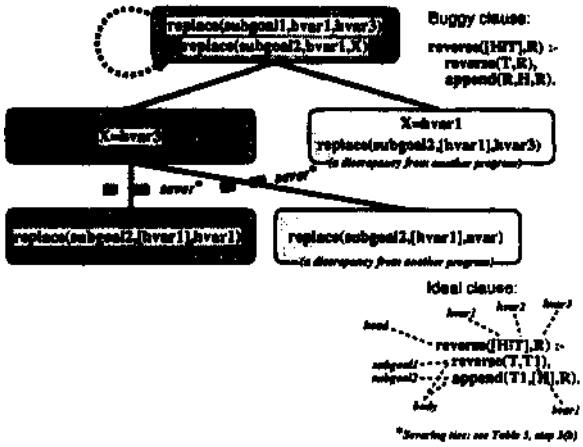

Figure 2: Direction of causality between discrepancies in a node

two discrepancies d_1 and *dl* are causally related and belong to the same concept/node, and if dl is an intersection generalization and d2, a variableization generalization, then *dl* causes (d2. This is because the intersection would be the discrepancy that is present together with all the various possible instantiations of the variableization. Thus, for example, students who are not confident in introducing variables in the body of a clause would omit the variable Tl in the recursive subgoal of reverse/2, and, as a result, use possibly different variables from the head (e.g., *R* or *H)* in place of Tl in the append/3 subgoal. Figure 2 illustrates.

Finally, the *sub concept-level* causality heuristic suggests that if two discrepancies *dl* and *dl* are causally related, and *dl* is a parent of d2, then *dl* is a probable cause of dl, i.e., based on empirical data, the child implies the parent. This is because a child is more likely to be encountered or seen with its parent, than the parent with a particular child. For instance, it is possible that the student in the example in Table 1 (and in Figure 1) put [Tl|H] in the head because he/she purposely omitted the append/3 subgoal in the body of his/her clause. This suggests that the student knew about the append/3 relation but decided that using [|] was better, at least in this case. However, it is more likely that the student omitted the append/3 subgoal as a result of putting [T1IH] in the head (Figure 3). This means that the student thought, incorrectly, that the [I] construct could be used to prepend a list to an object, and having dealt with the necessary concatenation, had no further need for a concatenation subgoal in the body of the clause.

3.3 A Similarity- and Causality-Based Clustering Algorithm

Existing approaches (e.g., [Lebowitz, 1986; Pazzani, 1993]) to using data and theory (causality) in concept formation use separate SBL and EBL components one after the other. In MMD, SBL and EBL are tightly coupled in the concept formation process. This entails two revisions to the basic algorithm presented in the previous section (rather than a separate algorithm altogether).

- Causal relationships are to be determined using the component-level causality heuristic.
- The directions of causalities are to be determined whenever possible using the concept and subconcept-level heuristics. This may lead to the severing of ties between a parent node and its child when the two are in fact unrelated.

These revisions are found in Table 3. Note that step (3b) effectively functions as a reorganization operator that is causality-based. Step (4) likewise reorganizes (prunes) the hierarchy, though it does this based on frequencies. Reorganization operators are especially important for incremental learners to mitigate ordering effects.

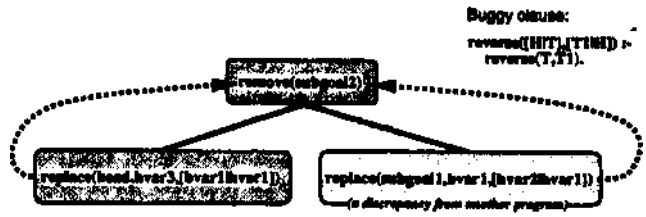

Figure 3: Direction of causality between discrepancies along a link

Table 3: Incorporating Causality into Concept Formation

1. Same as in Table 2, with the addition that causality relationships among discrepancies are to be determined using the component-level heuristic.
2. Same as in Table 2.
3. For every new node created in (2),
   (a) If concept-level causality exists among discrepancies in this node, record the direction of causality. If no such causality exists, retain the node nevertheless. (If the node needs to be split, it will be split by step (2) on another occasion.)
   (b) If subconcept-level causality exists between this node and its parent, record the direction of causality. If no such causality exists, sever the link between this child and its parent, linking it instead with its grandparent, then recheck for subconcept-level causality (i.e., step 3b).
4. Same as step (3) in Table 2.

4 Evaluation

For a preliminary empirical evaluation of the ability of MMD to discover actual misconceptions in real-world

behavior, a sufficiently large corpus of incorrect novice behavior, here in the form of buggy Prolog programs, had to be compiled. A total of 64 buggy reverse/2 and 56 buggy sumlist/2 programs (for the naive reversal of lists and for summing up the elements of a list of numbers, respectively) were obtained from third-year undergraduate students who have learned basic Prolog concepts, and then submitted for expert (teacher) analysis of the underlying misconceptions. The discrepancies between the buggy programs and their associated ideal programs were also computed and then fed into MMD. A point-by-point comparison of the misconception hierarchies generated by the expert and by MMD is presented elsewhere (see [Sison, Numao and Shimura, 1997]); here we compare instead the accuracies of the misconception/classification hierarchies generated by:

a) the basic UNIMEM-like similarity-based algorithm in Table 2[3] (called SMD in Figure 4); and

b) MMD

given *worst-case* orderings of the objects in the reverse/2 and sumlist/2 datasets. A misconception or classification generated by MMD or SMD is considered accurate if it matches that of the expert.

The results obtained are very encouraging (Figure 4). MMD was able to correctly classify most of the bugs in the student programs. The lower accuracy of the hierarchies generated by SMD were mainly due to incoherent groupings and multiple bugs, which SMD is insensitive to. The bugs which MMD (and of course SMD) was not able to classify correctly were primarily due to discrepancies which could be transformed to other, "more meaningful" discrepancies. For MMD to classify these bugs correctly, two options are possible. One option would be to give MMD the ability to recognize discrepancies between discrepancies (i.e., to transform one discrepancy to another). Alternatively, this task could be given to the preprocessor which computes discrepancies between buggy programs and an ideal. The second option is preferable since MMD's primary task is clustering discrepancies rather than transforming them.

5 Concluding Remarks

A similarity-based approach to misconception discovery is important because it reveals regularities in the data, which in turn may indicate the existence of underlying causalities. Moreover, in the absence of feature relar tionships deducible from the background knowledge, an SBL-generated node with high confidence can be learned as a new, though yet unexplainable, misconception. On the other hand, an explanation(causality)-based approach is necessary because concepts based solely on regularities might not be coherent and because some features in concept descriptions can be irrelevant. Furthermore, a similarity-based learner can only roughly classify an erroneous program but not specify the cause(s) of its errors.

[3]Using $\theta = \alpha = \beta = 1, \gamma \geq 0$.

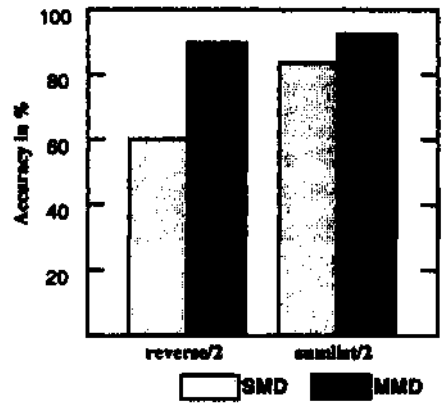

Figure 4: Accuracy of hierachies generated by SMD and MMD given *worst-case* orderings of objects in the reverse/2 and sumlist/2 datasets

The tight integration of similarity- and causality-based learning in the multistrategy unsupervised concept discovery system MMD has been shown to be useful, if not essential, for the the automatic construction of meaningful misconceptions that can be used to account for discrepant behavior in student programs. The applicability of MMD's approach should extend naturally to similar domains, i.e., domains in which causal relationships exist among components of behavior in the background knowledge. Future work will involve investigating mechanisms for generalizing misconceptions across problems, for using domain semantics to articulate causal relationships, and for exploiting and dynamically choosing other qualitative relationships (e.g., goals) that might exist in the background knowledge. MMD is a step toward the automatic discovery of (Prolog programming) misconceptions [Sison, 1997] and their use in multistrategic student modeling [Sison and Shimura, 1996b].

Acknowledgment

The first author thanks Ethel Chua Joy and Philip Chan for their assistance in compiling and analyzing the programs in the reverse/2 and sumlist/2 datasets.

References

[Barsalou, 1991] L. Barsalou. Deriving categories to achieve goals. *The Psychology of Learning and Motivation,* 27:1-64, 1991. .

[Corter and Gluck, 1992] J. Corter and M. Gluck. Explaining basic categories: Feature predictability and information. *Psychological Bulletin,* 111: 291-303, 1992.

[Fisher, 1987] D. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine learning.* 2:139-172, 1987.

[Flann and Dietterich, 1989] N. Flann and T. Dietterich. A study of explanation-based methods for inductive

learning, *Machine Learning,* 4:187-226, 1989.

[Gluck and Corter, 1985] M. Gluck and J. Corter. Information , uncertainty, and the utility of categories, In *Proceedings of the Annual Conference of the Cognitive Science Society,* pages 283-287. Lawrence Erlbaum, 1985.

[Komatsu, 1992] L. Komatsu. Recent views of conceptual structure, *Psychological Bulletin,* 112(3):500-526. 1992.

[Lebowitz, 1986] M. Lebowitz. Integrated learning: Controlling explanation, *Cognitive Science,* 10:219-240, 1986.

[Lebowitz, 1987] M. Lebowitz. Experiments with incremental concept formation. *Machine Learning,* 2:103-138, 1987.

[Martin and Billman, 1994] J. Martin and D. Billman. Acquiring and combining overlapping concepts. *Machine Learning,* 16:121-155, 1994.

[Michalski and Stepp, 1983] R. Michalski and R. Stepp. Learning from observation: Conceptual clustering. In R. Michalski, J. Carbonell, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach.* Tioga, Palo Alto, CA, 1983.

[Mooney and Ourston, 1989] R. Mooney and D. Ourston. Induction over the unexplained: Integrated learning of concepts with both explainable and conventional aspects, In *Proceedings of the Sixth International Workshop on Machine Learning,* pages 5-7. Morgan Kaufmann, 1989.

[Muggleton and Feng, 1990] S. Muggleton and C. Feng. Efficient induction of logic programs, In *Proceedings of the First Conference on Algorithmic Learning Theory,* Tokyo Ohmsha, 1990.

[Murphy and Medin, 1985] G. Murphy and D. Medin. The role of theories in conceptual coherence, *Psychological Review,* 92(3):289-316, 1985.

[Pazzani, 1993] M. Pazzani. Learning causal patterns: Making a transition from data-driven to theory-driven learning, *Machine Learning,* 11 (2/3): 173-194, 1993.

[Plotkin, 1970] G. Plotkin. A node on inductive generalization, *Machine Intelligence,* 5:153-163, 1970.

[Rips and Collins, 1993] L. Rips and A. Collins. Categories and resemblance, *Journal of Experimental Psychology: General,* 122(4):468-486, 1993.

[Sison, 1997]. R. Sison. Toward the automatic discovery of misconceptions. To appear in *Proceedings of the International Joint Conference on Artificial Intelligence,* 1997.

[Sison and Shimura, 1996a] R. Sison and M. Shimura. Incremental clustering of relational descriptions. Technical Report TR.96-0011. Department of Computer Science, Tokyo Institute of Technology, 1996.

[Sison and Shimura, 1996b] R. Sison and M. Shimura. The application of machine learning to student modeling: Toward a multistrategic learning student modeling system. In *Proceedings of the European Conference on Artificial Intelligence in Education,* pages 87-93, 1996.

[Sison, Numao and Shimura, 1997] R. Sison, M. Numao and M. Shimura. Discovering misconceptions using multistrategic conceptual clustering. To appear in *Proceedings of the World Conference on Artificial Intelligence in Education,* 1997.

[Stepp and Michalski, 1986] R. Stepp and R. Michalski. Conceptual clustering of structured concepts: A goal-oriented approach, *Artificial Intelligence,* 28:43-69, 1986.

[Thompson and Langley, 1991] K. Thompson and P. Langley. Concept formation in structured domains. In D. Fisher, M. Pazzani and P. Langley, editors, *Concept Formation: Knowledge and Experience in Unsupervised Learning.* Morgan Kaufmann, 1991.

[Tversky, 1977] A. Tversky. Features of similarity, *Psychological Review,* 84(4):327-352, 1977.

[Wisniewski and Medin, 1994] E. Wisniewski and D. Medin. On the interaction of theory and data in concept learning, *Cognitive Science,* 18:221-281, 1994.

[Yoo and Fisher, 1991] J. Yoo and D. Fisher. Concept formation over problem-solving experience. In D. Fisher, M. Pazzani and P. Langley, editors, *Concept Formation: Knowledge and Experience in Unsupervised Learning.* Morgan Kaufmann, 1991.