# On Using Theory and Data in Misconception Discovery

**Raymund Sison, Masayuki Numao,** and **Masamichi Shimura**
Department of Computer Science
Tokyo Institute of Technology
2-12-1 Ohokayama, Meguro-ku
Tokyo 152, Japan
{SISON}@CS.TITECH.AC.JP

**Abstract**

Approaches to concept formation tend to rely solely on similarities in the data, with the few that take into consideration causalities in the background knowledge doing so prior to or upon completion of a similarity-based learning phase. In this paper, we examine a multistrategic approach to misconception discovery that utilizes data and theory in a more tightly coupled way.

## Introduction

Most conceptual clustering systems for the unsupervised formation of concepts in Artificial Intelligence (AI) tend to rely solely on similarities in the data, a tendency that likewise characterizes much of concept learning research in cognitive psychology (Komatsu, 1992). Recently, however, the increase in the number of such works as those of Barsalou (1991), Rips and Collins (1993), and Wisniewski and Medin (1994) reveal an increasing dissatisfaction in cognitive psychology over similarity-based models' almost exclusive reliance on data and an increasing interest in the role of theories and goals in concept formation.

There are, to be sure, combined similarity-based (SB) and explanation-based (EB) AI learning systems that use data and theory to learn concepts, whether with or without supervision,[1] e.g., (Lebowitz, 1986; Pazzani, 1993; Flann & Dietterich, 1989; Mooney & Ourston, 1989; and Yoo & Fisher, 1991). Wisniewski and Medin (1994), however, noting that these systems treat SB and EB learning as phases that are performed one after the other, argue cogently that such loosely coupled approaches to using data and theory, while undoubtedly useful, remain inadequate as models of concept formation. This inadequacy becomes even more pronounced when dealing with *misconceptions*.

In general terms, a *misconception* is an incorrect understanding of a concept or procedure that results in systematic *discrepancies* in *behavior* (e.g., bugs in a program). These discrepancies can be expressed as *relational descriptions* — logic formulas that describe specific relations (i.e., the discrepancies) between a given behavior and an ideal one. Figure 1 illustrates.

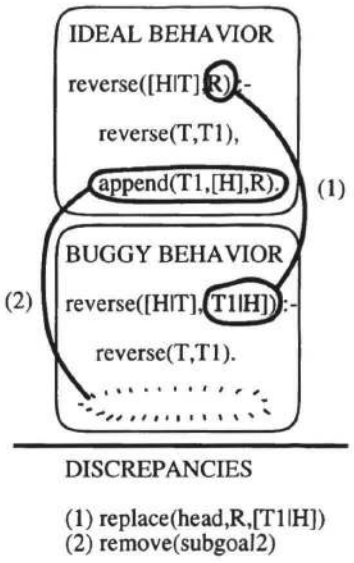


Figure 1: Discrepancies in behavior

The figure shows an ideal behavior in the form of the correct clause of a PROLOG program for reversing the order of a list of elements. The clause has a *head*, `reverse([H|T],R)`, which states that the reverse of a list that is made up of a first element, `H`, called the list's *head*, and a sublist, `T`, called the list's *tail*, is `R`. `R` is computed in the clause's *body*, which has two subgoals. The first subgoal, `reverse(T,T1)`, states that the reverse of the list `T` (recall that `T` is the tail of the list being reversed) is `T1`. The second subgoal, `append(T1,[H],R)`, states that `R` is just the concatenation of the list `T1` (which, according to the previous subgoal is the reverse of the tail of the list to be reversed) and the element `H`. In short, the clause as a whole states that the reverse of a list is the concatenation of the reverse of its tail and its head.

Below the correct clause in the figure is a clause written by a student. The student's clause differs from the ideal one in two ways. First, the student's clause has `[T1|H]` in the head instead of `R`. Second, it has only one subgoal, that for reversing the tail `T`. These two discrep-

[1] Supervision means supplying the learner with information (called labels) about the class or concept to which an object or event belongs. Thus, the supervised learner's task is to formulate a correct characterization of a given concept or set of concepts. In unsupervised learning, which is this paper's concern, objects are unlabeled and so the learner's task involves determining the concepts that exist among the objects as well as characterizing these concepts.

ancies are listed in relational logic form in the bottom of the figure. Knowledge about the misconception that caused such discrepancies can improve student remediation and lesson presentation. This paper examines an approach toward the automatic discovery of such misconceptions.

In the rest of the paper, we first present a similarity-based algorithm for clustering relational descriptions. Next we describe how causal relationships in the background knowledge can be exploited to construct or correct misconceptions while they are being formed. Finally we report some experimental results that show that the multistrategic approach to concept formation described in this paper enables the automatic construction of meaningful misconceptions from theory and data.

## Using Similarities in the Data to Form Concepts from Relational Descriptions

### The Basic Similarity Measure

The basis of our similarity-based algorithm is Tversky's (1977) contrast model:[2]

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

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

The features that are dealt with in this paper — behavioral discrepancies — are expressed as relational (rather than attribute-value) descriptions.[3] 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 & Feng, 1990) of two such descriptions.

### The Basic Similarity-based Relational Clustering Algorithm

Our similarity-based algorithm for clustering relational descriptions is incremental, so it takes one set of discrepancies at a time and classifies this object recursively into the nodes in a growing hierarchy that match it to a certain degree. Each node in the hierarchy denotes a concept (i.e., misconception), which is either (a) a generalization (intersection or variableization) of the subconcepts below it, or (b) a record of an instance, or both. Table 1 describes the basic algorithm. Further details can be found in (Sison & Shimura, 1996a), where the algorithm is called RC.

[2] All of the similarity-based views of concept learning adopt or assume some variant of this model (Komatsu, 1992).

[3] Attribute-value descriptions such as `height=tall` or `color=blue` can be used to express discrepancies only with difficulty.

Table 1: Basic procedure for clustering relational descriptions

1. From the children of a given node $N$ of a concept hierarchy, determine those that *match* the object $O$ (set of input discrepancies). The *match* function computes for every child node the set of commonalities, *Com*, and the degree of similarity, *Sim*, between this node and the new object, and determines whether *Sim* exceeds a system threshold, $\gamma$.
2. If no match is found, place $O$ under $N$. Otherwise, place $O$ in its appropriate position vis-a-vis the matching child(ren) of $N$. (This will involve increasing weight counters, creating new nodes, or further recursive clustering against child nodes.)
3. Nodes whose (*weight* · *height*) values fall below a system parameter may be discarded on a regular or demand basis.

The algorithm in Table 1, which we here call SMD, is similar to UNIMEM (Lebowitz, 1987) and COBWEB (Fisher, 1987), which are also incremental conceptual clusterers. UNIMEM's similarity measure, however, considers only the difference 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 uses a probabilistic (rather than set theoretic) concept representation and a corresponding probabilistic similarity measure (category utility (Gluck & Corter, 1985; Corter & Gluck, 1992)), and can therefore only produce disjoint clusters (but see the probabilistic clusterer in (Martin & Billman, 1994)). In this paper's context, disjoint clusters imply that the set of bugs in a particular behavior can only be classified under one "misconception," though there may well be several. Both UNIMEM and COBWEB deal only with attribute-value (rather than relational) descriptions (but see the COBWEB descendant in (Thompson & Langley, 1991)).

## Using Causalities in the Background Knowledge to Strengthen the Coherence of Concept Descriptions and Explain Discrepancies

### Causalities in the Background Knowledge

Similarity-based clusterers form categories on the basis of regularities (e.g., frequency, co-occurrence) among features in the data, but largely ignore qualitative relationships among these same features. We argue, however, that the presence of qualitative, particularly causal relationships between features of a concept are important in that they strengthen the coherence of a conceptual description (thus, e.g., their absence can warrant the splitting of a concept or an object when some regularities are coincidental), and they explain the regularities in the data. The latter, particularly the knowledge of causative features, is especially important when remedi-

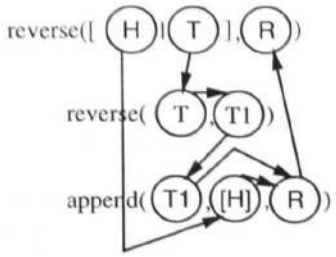


Figure 2: Causal relationships in the ideal behavior

ating or otherwise dealing with learner misconceptions.

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. Once the causative features have been determined, one can link 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.

**Example** To illustrate, recall the ideal PROLOG clause in Figure 1 for reversing a list. Said clause can be viewed as describing relationships, as desribed in the introduction, among four objects `H`, `T`, `T1`, and `R` used in the head and the two subgoals. Figure 2 now illustrates these relationships.

According to the component-level causality heuristic, the discrepancies in the said example are causally related since the `R` in the second discrepancy causes or enables that in the first (Figure 3). In other words, the student's use of the PROLOG list operator `[|]` in the head of his/her clause is related to the absence of the **append** subgoal in the body of his/her clause. The component-level causality heuristic, however, does not say anything about the direction of causality.

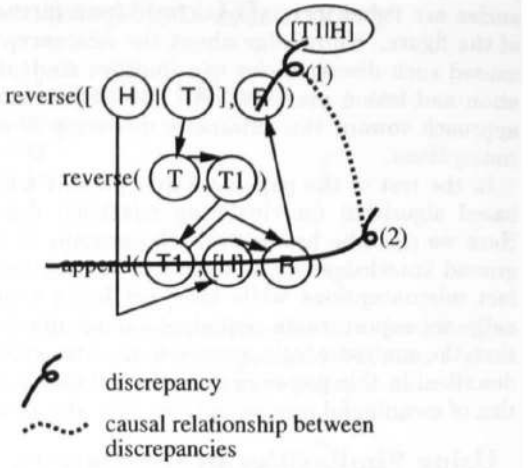


Figure 3: Causal relationships between discrepancies in behavior

Now if, in the misconception hierarchy, discrepancy (1) in Figure 1 (and in Figure 3) happens to occur under discrepancy (2),[4] then according to the subconcept-level causality heuristic, the former causes the latter. In other words, the student can be understood to have omitted the **append/3** subgoal as a result of his/her putting `[T1|H]` in the head. This means that the student thought, incorrectly, that the `[|]` 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 his/her clause.

## A Similarity- and Causality-Based Clustering Algorithm

Existing approaches (e.g., (Lebowitz, 1986; Pazzani, 1993) to using data and causality in concept formation use separate SBL and EBL components. In MMD, SBL and EBL are tightly coupled in the concept formation process. This entails two revisions to the basic algorithm (rather than an algorithm separate from that) in Table 1. First, causal relationships are to be determined using the component-level causality heuristic. Second, 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 2, which shows the basic similarity- and causality-based algorithm, called MMD for multistrategy misconception discovery. MMD and the causality heuristics above are explained in more detail in (Sison, Numao & Shimura, 1997).

[4] Which is indeed the case with the data we have gathered for and used in our experiments.

Table 2: Basic procedure for similarity- and causality-based misconception discovery

1. Same as in Table 1, with the addition that causality relationships among discrepancies are to be determined using the component-level heuristic.
2. Same as in Table 1.
3. For every new node created in (2), determine and record the existence of concept- and subconcept-level causalities. If no concept-level causality exists among discrepancies in this node, retain the node nevertheless. If no subconcept-level causality exists between this node and its parent, sever the link between this child and its parent, and promote it upwards.
4. Same as step (3) in Table 1.

## Experiment

### Experimental Method

In this paper, we look at the effect of varying the parameters of SMD and compare the performance of these SMD variants against that of MMD. The data we use are 64 buggy `reverse/2` programs obtained from third-year undergraduate students who have learned basic PROLOG concepts. These programs were submitted for expert (teacher) analysis of their underlying misconceptions. The discrepancies between the buggy programs and their associated ideal programs were also computed and then fed, in worst-case order,[5] into several variants of SMD (each variant having different values for its parameters) and into MMD.[6] A misconception or classification generated by MMD or SMD is considered accurate if it matches that of the expert.

Varying the parameters of SMD, particularly the parameters of the $Sim$ function it uses, reflects various similarity models. For example, setting $\theta$ to 1 and $\alpha$ and $\beta$ to 0 produces Restle's (1961) model of similarity. The reverse, i.e., setting $\theta$ to 0 and $\alpha$ and $\beta$ to 1 yields Restle's (1961) model of psychological distance, which is basically what UNIMEM uses. Setting $\alpha$ and $\beta$ to fractional values (when $\theta$ is 1) is sometimes useful, as Weber (1996) shows in his particular domain. These variants are summarized in Table 3, and the results of the experiment are shown in Figure 4.

### Discussion of Results

Figure 4 shows that MMD was able to correctly identify most (92%) of the misconceptions in the buggy programs that the expert also could. The figure also clearly indicates that exploiting causal relationships in the background knowledge improves the classification performance of a similarity-based learner. What the figure does not reveal is that the classifications generated by all the SMD variants do not contain information regarding causation and can therefore hardly be considered as misconceptions.

[5] Since the algorithms are incremental, the order in which the e discrepancy sets are presented to the algorithms can affect the accuracy of the resulting hierarchies. A worst-case ordering is one which maximizes error in a hierarchy.

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

Table 3: Some variants of SMD

| *Model* | $\theta$ | $\alpha$ | $\beta$ | $\gamma$ | *Name in Figure 4* |
|---|---|---|---|---|---|
| RC | 1 | 1 | 1 | 0 | SMD4 |
| Restle-1 | 1 | 0 | 0 | 1 | SMD1 |
| Restle-2 | 0 | 1 | 1 | -1 | SMD3 |
| Weber | 1 | .2 | .4 | 0 | SMD2 |

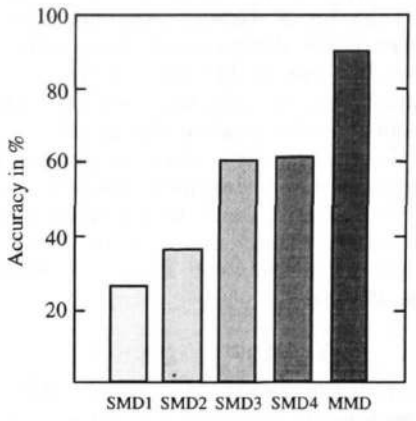


Figure 4: Accuracy of hierarchies generated by MMD and the SMD variants in Table 3

The relatively lower accuracy of the hierarchies generated by SMD were mainly due to incoherent groupings and multiple bugs, which SMD is insensitive to. This insensitivity seems to have become more pronounced in the SMD variants (SMD1, SMD2) that assigned weights lesser than 1 to feature dissimilarities. This is not surprising since the presence of dissimilarities between two similar buggy behaviors may indicate the existence of more than one bug.

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.

## Conclusion

A similarity-based approach to misconception discovery is important because it detects regularities in the data, which in turn may indicate the existence of underlying causalities. On the other hand, an explanation(causality)-based approach is necessary be-

cause concepts based solely on regularities might not be coherent. Furthermore, a similarity-based learner can only roughly classify an erroneous program but not specify the cause(s) of its errors.

The 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 presence of qualitative, particularly causal relationships between features of a concept enable the splitting of an object with multiple bugs, thereby increasing the hierarchy's accuracy, and provides a causal explanation the regularities in the data. The latter is especially important when remediating or otherwise dealing with learner misconceptions. MMD is a step toward the automatic discovery of (PROLOG programming) misconceptions (Sison, 1997) and their use in multistrategic student modeling (Sison & Shimura, 1996b).

## Acknowledgment

The first author thanks Ethel Chua Joy and Philip Chan for their assistance in compiling and analyzing the buggy PROLOG programs.

## References

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

Corter, J. & Gluck, M. (1992). Explaining basic categories: Feature oredictability and information. *Psychological Bulletin,* 111, 291-303.

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine learning.* 2, 139-172.

Flann, N & Dietterich, T. (1989). A study of explanation-based methods for inductive learning. *Machine Learning,* 4, 187-226.

Gluck, M & Corter, J. (1985). Information , uncertainty, and the utility of categories. *Proceedings of the Annual Conference of the Cognitive Science Society,* (pp 283-287). Hillsdale, NJ: Lawrence Erlbaum.

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

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

Lebowitz, M. (1987). Experiments with incremental concept formation. *Machine Learning,* 2, 103-138.

Martin, J. & Billman, D. (1994). Acquiring and combining overlapping concepts. *Machine Learning,* 16, 121-155.

Michalski, R & Stepp, R. (1983). Learning from observation: Conceptual clustering. In R. Michalski, J. Carbonell, & T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach.* Palo Alto, CA: Tioga.

Mooney, R & Ourston, D. (1989). Induction over the unexplained: Integrated learning of concepts with both explainable and conventional aspects. *Proceedings of the Sixth International Workshop on Machine Learning,* (pp. 5-7). San Mateo, CA: Morgan Kaufmann.

Muggleton, S & Feng, C. (1990). Efficient induction of logic programs. *Proceedings of the First Conference on Algorithmic Learning Theory.* Tokyo: Ohmsha.

Murphy, G & Medin, D. (1985). The role of theories in conceptual coherence. *Psychological Review,* 92, 289-316.

Pazzani, M. (1991). A computational theory of learning causal relationships. *Cognitive Science,* 15, 401-424.

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

Plotkin, G. (1970). A node on inductive generalization. *Machine Intelligence,* 5, 153-163.

Restle, F. (1961). *Psychology of Judgment and Choice.* New York: Wiley.

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

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

Sison, R. & Shimura, M. (1996a). Incremental clustering of relational descriptions. Technical Report TR96-0011. Department of Computer Science, Tokyo Institute of Technology.

Sison, R & Shimura, M. (1996b). The application of machine learning to student modeling: Toward a multistrategic learning student modeling system. *Proceedings of the European Conference on Artificial Intelligence in Education,* (pp. 87-93).

Sison, R., Numao, M. & Shimura, M. (1997). Using data and theory in multistrategy (mis)concept(ion) discovery. To appear in *Proceedings of the International Joint Conference on Artificial Intelligence.*

Stepp, R & Michalski, R. (1986). Conceptual clustering of structured concepts: A goal-oriented approach. *Artificial Intelligence,* 28, 43-69.

Thompson, K & Langley, P. (1991). Concept formation in structured domains. In D. Fisher, M. Pazzani & P. Langley (Eds.), *Concept Formation: Knowledge and Experience in Unsupervised Learning.* San Mateo, CA: Morgan Kaufmann.

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

Weber, G. (1996). Episodic learner modeling. *Cognitive Science,* 20, 195-236.

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

Yoo, J & Fisher, D. (1991). Concept formation over problem-solving experience. In D. Fisher, M. Pazzani & P. Langley (Eds.), *Concept Formation: Knowledge and Experience in Unsupervised Learning.* San Mateo, CA: Morgan Kaufmann.