Diagnosing Errors in Statistical Problem-Solving: Associative Problem Recognition and Plan-Based Error Detection

Marc M. Sebrechts
Lael J. Schooler
Cognitive Science Program & Department of Psychology
Wesleyan University

Abstract

This paper describes our model for diagnosis of student errors in statistical problem-solving. A simulation of that diagnosis, GIDE, is presented together with empirical validation on student solutions. The model consists of two components. An "intention-based" diagnostic component analyzes solutions and locates errors by trying to synthesize student solutions from knowledge about the goal structure of the problem and related knowledge about planning errors. This approach can account for about 82% of the lines and over 95% of the goals in a set of 60 student t-tests. When solutions contain errors in procedural implementation such plan-based analysis is quite effective. In many cases, however, students do not pursue an "appropriate" solution path. The diagnostic model, therefore, includes a second component which is used to determine which type of problem the student is using: it is modeled by a spreading activation network of statistical knowledge. On a sample of 38 student solutions, the simulation correctly identified 86% of the problem types. The model appears to account for a wide range of problem-solving behavior within the domain studied. The preliminary performance data suggest that our model may serve as a useful part of an intelligent tutoring system.

Introduction

An important part of problem-solving is the ability to detect and explain errors in proposed solutions. Such "diagnostic" ability has gained particular prominence from attempts to analyze faulty problem-solving during nuclear power plant failures. However, a similar type of skill constitutes an important part of the daily activity in almost any classroom. In both settings, the assumption is that we can learn by our failures.

Although the behavior is common, it is not simple. Diagnosis includes a wide range of general strategies and domain specific behaviors. In this paper, we report our initial attempts to analyze one common type of instructional diagnosis: locating and explaining errors in solutions to statistics problems.

We will first describe the general principles that guide our conception of the task. They are based on both our informal analysis of expert protocols in statistics and on reported strategies in the literature. These principles then serve as the basis for our model of diagnosis. That model is made explicit in a simulation. Finally, we report tests of our model on data collected from students. These analyses are meant to provide a test of sufficiency only. Additional data will be needed to verify a more detailed mapping between the "process" of our implementation and that of experts.

General Principles

Before describing the diagnostic system, it should be noted that "diagnosis" covers a wide range of problems. Its most familiar use is in medicine, and substantial advances have been made in developing a model of medical diagnosis (Clancey, 1985; Clancey and Lestinger, 1984; Shortliffe, 1976). Although this research borrows heavily from the medical domain, the "diagnostic" goals are rather different. In medicine, diagnosis is used to determine the underlying cause of a set of symptoms. In the case of statistics (or other situations where we are trying to instruct the student), the ultimate diagnostic goal is not only to understand the source of the error in the solution, but to specify the source of the error *with respect to the student's conception of the problem.*

Our model of this type of diagnosis is guided by a series of general principles about the nature of errors. First, it is recognized that errors in problem-solving are frequently systematic rather than random. Even for an apparently simple domain such as subtraction, there are large numbers of potential systematic errors. Friend and Burton (1981), for example, listed 110 'primitive' subtraction errors which can lead to a much larger number of compound errors (Brown & Burton, 1978; Burton, 1982). Treating error diagnosis as a problem of detecting misdirected systematicity makes it possible to develop useful strategies for isolating and correcting the mistakes. The model of the expert diagnostician must therefore include systematic deviations from correct procedures as part of its knowledge base.

Second, an error is, at least in part, generated by its context. It is often difficult to say whether or not a particular action was appropriate or inappropriate without knowing the reason for its inclusion. The prototype case for such reasoning is programming. There are an infinite number of syntactically correct steps that are possible in even a moderately complex program. However, the number of legitimate steps for a particular program are far more constrained.

In the more general case of problem-solving, it is often difficult to judge the correctness of a component procedure in isolation. It is therefore necessary to embed knowledge of errors within a context that characterizes the normal structure of problem solutions: and the knowledge of the diagnostician must somehow capture that embedded structure. Below it is argued that such a context must include a problem description and associated goal structure in order to capture the "intention" of the student.

The third general principle is closely related to the second: solution strategies share many concepts in common. This means that solution components can be viewed as part of complex heterarchy. Detecting individual concepts or formulae will not be sufficient to analyze a solution. With respect to our model, this principle argues for a process of search that is goal and plan directed. It is the goal/plan structure that disambiguates the role of a procedure. It is important to recognize that this model attempts to capture "diagnosis" by an instructor; that process need not, and usually does not, mimic student problem solving. There is evidence that an explicit plan structure can improve students' performance in programming (Miller, 1978), but that data actually suggests that students do not usually follow such systematic planning in their unconstrained problem-solving.

The Intention-Based Approach

These principles concerning the relationship of a knowledge base and error diagnosis served as the basis for our model of diagnosis in statistics (Sebrechts, Schooler & LaClaire, 1986; Sebrechts, LaClaire, Schooler & Soloway, 1986). This portion of the model is based on a strategy developed by Johnson and Soloway (1985) in the realm of programming (PROUST). This approach is called "intention-based" diagnosis, since it uses a goal structure that represents the problem the student apparently "intended" to solve. In this approach a problem is described in terms of a series of necessary and sufficient goals. The goals in turn have a variety of plan implementations, both correct and "buggy" that can be used to describe the student's solution. The instructors' knowledge is thus described as consisting of knowledge about the domain and knowledge about common errors. In addition, errors are thought to be understood within a context of goals and plans. A statement's "meaning" is determined relative to the broader solution context.

The simulation for statistics, called GIDE, analyzes a solution by attempting to describe how the student solution matches, or fails to match, some realization of an intended solution. The procedure consists of analysis-by-synthesis: an attempt is made to analyze a solution by building a plausible description of how that solution satisfies the goals defined by the problem.

For each problem, GIDE has a problem description that indicates those goals that must be satisfied by any correct solution. GIDE attempts to find some satisfaction of those goals in the student's solution by testing a series of plans. If no correct solution is located, GIDE tries to explain the discrepancy. This can be done by trying "buggy" plans that represent common conceptual errors in a plan (such as confusing standard-error and standard-deviation), or by examining "buggy" rules representing common mistakes across different types of plans (such as reversing a sign in a computation).

GIDE attempts to extend the model used for programming in PROUST to different domains. Student solutions in statistics differ in numerous ways from programming solutions. First, students frequently leave out steps. Components that are "obvious" or can be calculated mentally are left out. In addition, there is a very loose syntax, which allows frequent changes in construction. In contrast to most programming languages, students use assignment to values, they include free (unbound) expressions in their solutions, and they occasionally change the symbols they are using. These are all components that can be analyzed successfully by teaching assistants.

In order to handle the wide range of expressions used, GIDE can match a procedure in several ways. The first strategy is to look for an appropriate symbolic structure. If a student writes "StdE = 20 / sqrt n", GIDE will match this to its internal representation of one form for standard error (?Se = ?Sd / sqrt ?count). Another way in which the student may indicate the standard error is by using the appropriate values. The form "5 = 20 / 4" ,would also be recognized as standard error *if* ?Sd had been previously bound to 20 and ?count had been bound to 16. Finally, if a plan is not satisfied symbolically or by value, GIDE will attempt to find a free expression. Thus, for example, the "-14" on line 9 of Figure 2, is recognized as a deviation score.

In order to handle steps that are left out, GIDE uses "implicit matching". GIDE contains a dependency tree that shows the relationship among knowledge of concepts: for example, determining the variance requires the sum-of-squares. If the student includes in their solution the correct variance, but not the sum-of-squares, GIDE infers that the student understands the sum-of-squares. The details of these techniques are described in Sebrechts, Schooler and LaClaire (1986). Here we will focus on the empirical outcome of this general approach.

Empirical Tests of Plan Analysis

In order to evaluate our diagnostic model, we examined GIDE's performance on a series of 43 solutions collected from students in an introductory statistics course. In order to examine the types of plans needed for different solution strategies we used two testing conditions. In the first condition (called "Directed") the problem was accompanied by a set of directions indicating the appropriate steps for solution, and was similar to a homework assignment. In the second condition ("Undirected") another problem was given as an exam problem without any accompanying materials.

The specific statistical problem we have examined is a repeated-measures t-test. This type of problem attempts to determine whether or not there is a reliable difference between two measurements on the same group. For example, you might use this test to determine if a training program has improved the efficiency of workers by measuring efficiency before and after training (Figure 1).

Student solutions were written on paper and were coded directly into GIDE. An example solution from each of the two conditions is shown in Figures 2 and 3. The only information added by GIDE are line numbers for reference. GIDE's output for the two solutions is included with each solution.

Buggy Solutions. Not surprisingly, a greater percentage of solutions had bugs in the "undirected" condition. Of those solutions with bugs, however, the greatest percentage in both conditions consisted of missing and implicit goals. An implicit goal is one that is not explicitly stated but is obvious from other concepts. For example, in order to compute an average, the student must know the sum. GIDE catches over 90% of such implicit goals based on its knowledge of

Problem Statement:

An employer at an automotive plant is interested in determining whether or not a training program can improve efficiency of his employees. He has tried it out tentatively for four work groups in one of the plants. Below are the results of the trial: efficiency is measured by a standardized test ranging from 1 (extremely efficient) to 25 (extremely inefficient). Determine whether or not there is a reliable change in efficiency.

Work Group	Pre-Training	Post-Training	Post Minus Pre
A	14	9	-5
B	19	7	-12
C	16	10	-6
D	18	5	-13

Figure 1. An example of a repeated-measures t-test.

dependencies. Missing goals are those that never appear in the solution and are not implicit. GIDE catches a moderately high number of such goals (71% and 86% for directed and undirected problems respectively). The main reason why performance is not higher is that the mechanism for implicit matching is too powerful. It occasionally gives credit for knowing a concept that is actually not evident in the student solution.

GIDE was able to detect all of the relatively small number of specific plan errors. There were, however, important differences between the types of errors generated in the two conditions, as can be seen in Figure 4. For Directed problems, the errors tended to be more low level, involving failures in computational plans. In the Undirected case, students tended to make higher level mistakes. For example, in Figure 3, the student confused the standard deviation with the standard error.

Problem-Recognition Through an Associative Network

GIDE performed extremely well on the set of problems initially provided. As we began to extend the domain of GIDE's capabilities, however, we discovered that the system could become involved in extensive searches that were not reflected in our protocols of experts. When we tested experts on these problems, they would say, "this student is using the wrong kind of test", before doing a careful analysis. In brief, there was a problem recognition phase that was not captured by the systematic approach of GIDE.

Of course, GIDE can manage to get around this difficulty by an exhaustive search. However, such a search would eliminate the heuristic value of intention-based diagnosis. As a consequence it

Directed Student Solution (id #48)

```
 1:      -5
 2:     -12
 3:      -6
 4:     -13
 5:       n
 6:       4
 7:   -6 + -13 + -5 + -12 = -36
 8:    d_dbar
 9:     -14
10:     -21
11:     -15
12:     -22
13:    d_dbar2
14:     196
15:     441
16:     225
17:     484
18:    1346
19:  1346/3 = 448 = s2
20:  sqrt 448 = 21 = s
21:  sdbar = 21/2 = 11
22:   -9/11 = t
```

The average was implicitly included in your solution.

Your calculation of the deviations was incorrect. A useful way to avoid this mistake is to note that the sum of the deviations should always equal 0. The sum of your calculated deviations equals 72.0

In the process of calculating the standard deviation, you added the individual deviation scores to the mean of the deviations: instead, you should have subtracted the mean of the deviations from the individual deviation scores. For example, where you have -14, you should have 4.0.

These erroneous values were used to complete the analysis.

Figure 2: Student Solution (id #48) and GIDE's associated comments.

Undirected Student Solution (id #2)

```
 1:      d
 2:     -4
 3:     -8
 4:     -4
 5:     -4
 6:   -20 = sum d
 7:    x_dbar
 8:      1
 9:      3
10:      1
11:      1
12:    x_dbar2
13:      1
14:      9
15:      1
16:      1
17:    ss = 12
18:    dbar = -5
19:  t = (dbar   Uo)/ sdbar
20:  Sd = sqrt (12/3) = sqrt 4 = 2
21:  sdbar = sd/sqrt n = 2/sqrt 4 = 1
22:  t = -5/2 = -2.5
23:    alpha = 0.05
24:    tcrit = 3.18
```

The count of the observations was implicitly included in your solution.

The variance was implicitly included in your solution.

On line 22, when calculating the t statistic, you substituted the standard deviation for the standard error.

This erroneous value was used to complete the analysis.

Figure 3: Student solution (id #2) and GIDE's associated comments.

GIDE's Performance

Criteria	Directed		Undirected	
Solutions				
Total number of solutions	43		17	
Total receiving complete analysis	43	(100.0%)	16	(94.0%)
Total receiving partial analysis	0	(0.0%)	1	(6.0%)
Goals				
Total number of goals in solutions	445		216	
Total number of goals correctly analyzed	443	(99.5%)	207	(96.0%)
Total number of goals incorrectly analyzed	2	(0.5%)	9	(4.0%)
Lines				
Total number of lines included in solutions	1203		403	
Total number of lines correctly interpreted	986	(82.0%)	329	(81.6%)
Total number of lines misinterpreted	1	(0.0%)	9	(2.2%)
Total number of lines unaccounted for	216	(18.0%)	65	(16.1%)
Bugs				
In solutions:				
total number of solutions with bugs	27	(62.8%)	17	(100.0%)
total number of bugs	69		91	
missing goals	28	(40.5%)	37	(40.6%)
implicit goals	35	(50.7%)	48	(52.7%)
plan errors	6	(8.6%)	5	(5.5%)
errorneous solutions	0	(0.0%)	1	(1.1%)
Detected:				
total number of bugs detected	58	(84.0%)	84	(92.0%)
missing goals	20	(71.4%)	32	(86.5%)
implicit goals	32	(91.4%)	47	(97.9%)
plan errors	6	(100.0%)	5	(100.0%)
erroneous solutions	0		1	(100.0%)
false alarms	8		4	

Figure 4: Summary of GIDE's performance on directed and undirected student solutions.

would require a knowledge base and search times that are unreasonable as parts of a psychological model of the task.

In many cases, although the problem description does suggest what the student is trying to do, it does not capture the strategy that the student selects toward that end. Experts appear to be able to recognize how statements are related to more global problems. This was demonstrated in the case of programming by Adelson (1981). She found that although novices grouped program statements syntactically, experts grouped statements according to the programs from which they were derived. In terms of our diagnostic problem, this implies that experts should be able to recognize the type of solution being presented by examining constituent statements.

This is consistent with the problem analysis we have observed in statistics. The instructor tends to classify solutions fairly quickly, frequently with reference to specific aspects of the solution. Our model of how this occurs is that the diagnostician scans through the problem, identifying the component terms of the solution. Each term provides some degree of activation for associated concepts. The problem type that receives the highest level of activation during scanning is selected as the appropriate response.

We have simulated these processes by a network structure which consists of letters, symbols ("words"), goals, and problem types. The basic layer structure is similar in concept to that described by McClelland (1979) in his cascade model. Nodes at each level have excitatory links to nodes at the next higher level. In addition, there are reciprocal links from symbols ("words") to individual letters as in the McClelland and Rumelhart (1981) model, but in contrast to their model, there are no inhibitory links. (Likewise our model does not include any phonetic or morphemic information.)

The model is constrained by the fact that its only recognition capability is the particular statistical domain of interest. There are at least two plausible ways to characterize this network. It can be viewed as a subset of a more complete "reading" network in which the domain-relevant elements have higher initial activation levels; as such, only those initially activated nodes are relevant. Alternatively, it can be viewed as a domain-specific network for statistics which is activated more or less as a unit. The model does not make any differentiation between those options.

At the beginning of a particular problem type, GIDE's goal structure is used to construct the appropriate network. This process can be thought of as potentiating the portions of the network that are relevant to the statistics problems. Based on the problem description, each goal is linked to its subgoals. Thus, for example, the goal for Standard-Deviation would be linked to the subgoals of Mean, Count, Sum-of-Squares. The goals and subgoals are likewise linked to associated symbols. Standard-Deviation is linked to the symbol 'Std', a common abbreviation. Finally, each symbol is linked to its constituent letter nodes, which indicate both the character and the position of each letter. So, the letter node "s-" would indicate an initial 's' and would activate the "Std" symbol, whereas the letter node "-s" indicates a terminal 's' and would not directly activate 'Std'.

Once the network is established in this way, each of the symbols in the solution is "read" and decomposed into its constituent letters. The related symbols and goals are then activated through spreading activation following a reduced version of the general model described by Anderson (1983a: 1983b) for ACT*. There are no direct connections between letters, but one letter can indirectly activate another letter through a symbol or goal. The network is run as though activation occurred in parallel. Following the spread on each cycle, all elements are decayed. This results in a dynamic set of activation levels which are updated as the lines are "scanned."

This network approach to recognition makes sense in light of behaviors we have observed. Statistics includes a set of fairly common symbols. However, unlike the case of reading English in which words and non-words are differentiated, in statistics there are numerous deviations from those symbols, depending on idiosyncracies of individual students. Spreading activation helps to capture such deviations as well as to locate minor errors without having to anticipate all possible forms of each symbol. Thus, for example, "d-dbar" is an appropriate form for finding a deviation score in a repeated measures t-test. Some students use "x-dbar", which may indicate either a different notation or an error in the procedure. In either case, it does suggest that the student is getting deviation scores base on the mean of deviations ("dbar"). When the network is presented with "x-dbar" it will activate "d-dbar" given the strong similarity of the two symbolic expressions.

Empirical Tests of Network Activation Analysis

In order to evaluate this portion of the model, we conducted network activations on 38 student solutions. The network included four standard statistical tests that served as problem types: repeated-t, independent-t, repeated ANOVA, and independent ANOVA. We compared the highest goal activation level in the network with categorization by the two authors. On average, the system was able to identify 86% of the problems correctly. It was more successful at identifying repeated-measures t-tests (95%) than independent sample t-tests (75%). This is due to the fact that repeated measures t-tests in our sample are usually conducted using deviation scores: the terms associated with those deviations are more distinct that the terms in the independent t-test.

Informal analysis suggests that this performance is roughly comparable to that of a teaching assistant. More importantly, the problems that tend to create the greatest difficulty for instructors are also those that provide the least differentiation for the simulation. Likewise the simulation's microbehavior reflects sensible changes in the relative activations of different goals as lines are scanned. Figure 5 shows a trace of activation levels for four types of statistical test. Through symbol line 14, the system indicates roughly equal activation for repeated and independent t-tests. This is because most of the symbols are common to both tests. The presence of SUMX2 and XBAR2 has provided slightly greater activation for independent-t, since repeated-t is usually constructed with deviation scores rather than a second set of sums. DBAR (lines 20 and 25 in Figure 5), however, is the mean deviation score and is used only in repeated-t. As a consequence the final solution is judged to be a repeated t based on overall activation. Since these tests are in fact quite similar, and the student has used symbols common to both, as we would expect, the activation differences are not very large.

Activation Level Analysis of Student Solution # 16

Student Solution

1 d
2 4
3 8
4 4
5 4
6 sum x1 = 42
7 n1 = 4
8 xbar1 = 10.5
9 ss = 29
10 alpha 0.05
11 sumx2
12 n2 = 4
13 xbar2 = 5.5
14 ss = 5
15 n = 4
16 k = 2
17 alpha = .05
18 sum d = 20
19 nd = 4
20 dbar = 5
21 ss = 12
22 sd = sqrt (12/(n-1))
23 sdbar = sqrt (12/3) = 2
24 sdbar = 2/sqrt 4 = 1
25 t = (dbar-ud)/sdbar = (5-0)/1 = 5

Goal Activation Levels

Symbols in Solution	Repeated t-test	Independent t-test	Repeated ANOVA	Independent ANOVA
D	0.0	0.0	0.0	0.0
X1	0.036	0.035	0.0	0.0
N1	0.012	0.012	0.0	0.0
XBAR1	0.363	0.554	0.100	0.028
SS	0.188	0.378	0.021	0.405
ALPHA	0.112	0.144	0.004	0.338
SUMX2	0.029	0.760	0.000	0.048
N2	0.006	0.461	0.000	0.010
XBAR2	0.157	0.887	0.047	0.014
SS	0.112	0.471	0.010	0.351
N	0.072	0.302	0.006	0.224
K	0.491	0.638	0.004	0.144
ALPHA	0.230	0.254	0.001	0.291
D	0.230	0.254	0.001	0.291
ND	0.975	0.104	0.000	0.119
DBAR	0.873	0.604	0.092	0.041
SS	0.309	0.204	0.019	0.350
SD	0.989	0.123	0.004	0.297
N	0.633	0.079	0.003	0.190
SDBAR	0.955	0.484	0.090	0.065
SDBAR	1.640	0.877	0.143	0.066
T	3.471	2.982	0.092	0.042
DBAR	2.612	1.855	0.208	0.069

Figure 5: A trace of goal activation levels for a line-by-line evaluation of a student problem.

Initial problem recognition is good, but it should be noted that there are several assumptions implicit in these results. In this analysis, we have assumed that the highest activation level is the appropriate characterization of the goal selection. The reasoning behind this approach is that an expert actually scans a problem in a non-linear fashion. The problem-type is defined by focusing on the most salient features in the problem. In the simulation, salience is described by peak activation. Our analysis suggests that the simulation can handle student solutions in ways similar to that of an instructor. However, additional data will be needed to confirm the model at a process level.

Summary and Conclusions

We have presented two components of what we believe to be a reasonable characterization of diagnosis for statistics. The combination of "automatic" problem recognition and more deliberate goal-directed search is consistent with other theories in psychology. The general distinction has been part of psychological models for a number of years (Shiffrin & Shneider, 1977). Only recently, however, have these components been integrated into a model that attempts to account for problem solving (Hunt & Lansman, 1986). Despite substantial advances in the development of low level models of processing (McClelland, Rumelhart, and the PDP Group, 1986), there is still a case to be made for separating automatic associative processing from goal-based knowledge (Norman, 1986, calls this "deliberate conscious control").

This two component model also exhibits behavior that is comparable in many ways to that of instructors. The goal-direction, "intention-based" aspect provides a way to account for a range of errors. This approach provides analysis of over 80% of the individual lines and almost all of the goals in a set of t-tests that we have collected from students. Observation of instructors, however, indicates that they are able to select individual types of procedures by scanning the problem. We have modelled this behavior as the spreading activation of an associative network of letters, symbols ("words"), goals and problem types. Using highest level of activation as the criterion for goal selection, this method can correctly identify 86% of the problems tested. In order to serve as a good psychological model, difficulties encountered by the simulation should mirror those of instructors. Our preliminary observations confirm that match, although it will be necessary to verify the behavior with a larger sample of independent judges.

Although the model has met reasonable performance criteria, it is not an exhaustive model of diagnosis, and there are at least two important qualifications on the results. First, the two components of the model, automatic recognition and goal-based reasoning, are currently only weakly linked in the simulation. Our model of these components would suggest greater interaction between the components, in a manner similar to Hunt and Lansman's (1986) production-activation model. The activation levels available during problem-scanning should be used to provide more specific direction to the deliberate search. In the current simulation those levels are only used to select among problem types at the global level. Second, the data we have reported deal with a relatively circumscribed problem space in statistics; additional data will be needed to demonstrate the generality of the model.

The proposed model seems to provide a reasonable approximation to several important aspects of diagnosis. The "intention-based" component of the model has now been validated in both programming (Soloway & Ehrlich, 1984; Sack et al., in prep.) and in statistics. The spreading activation component has not been explicitly validated outside of statistics in its current form, but it has been shown to be of general utility as a model for different kinds of psychological processes (Anderson, 1983b). Extending the model we have described here should prove useful for the development of intelligent tutoring systems.

References

Adelson, B. (1981). Problem-solving and the development of abstract categories in programming languages. **Memory and Cognition**, *9*, 422-433.

Anderson, J.R. (1983a). **The architecture of cognition.** Cambridge, MA: Harvard University Press.

Anderson, J.R. (1983b). A spreading activation theory of memory. **Journal of Verbal Learning and Verbal Behavior**, *22*, 261-295.

Brown, J.S. & Burton, R.R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. **Cognitive Science**, *2*, 155-192.

Burton, R.R. (1982). Diagnosing bugs in a simple procedural skill. In D. Sleeman & J.S. Brown, (Eds), **Intelligent tutoring systems**. pp.157-183.

Clancey, W.J. (1985). **Acquiring, representing, and evaluating a competence model of diagnostic strategy**. Report No. STAN-CS-85- 1067, Department of Computer Science, Stanford University.

Clancey, W.J. and Lestinger, R. (1984). NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. In W.J. Clancey, and E.H. Shortliffe, (Eds.), **Readings in medical artificial intelligence: The first decade**. pp. 361-381.

Friend, J. & Burton, R.R. (1981). **Teacher's manual of subtraction bugs**. CIS Working Paper. Xerox Palo Alto Science Center.

Hunt, E. & Lansman, M. (1986). Unified model of attention and problem solving. **Psychological Review**, *93(4)*, 446-461.

Johnson, W.L. (1985). **Intention-based diagnosis of errors in novice programs**. Research Report #395. Department of Computer Science, Yale University, New Haven, CT.

Johnson, W.L. & Soloway, E. (1985). PROUST: An automatic debugger for Pascal programs. **Byte**. April, 179-190.

McClelland, J.L. (1979). On the time relations of mental processes: An examination of processes in cascade. **Psychological Review**, *86*, 287-290.

McClelland, J.L. & Rumelhart, D.E. (1981). An interactive model of context effects in letter perception: Part I. An account of basic findings. **Psychological Review**, *88*, 375-407.

McClelland, J.L., Rumelhart, D.E., & the PDP Research Group (1986). **Parallel distributed processing**. Cambridge: MIT Press.

Miller, M.L. (1978). A structured planning and debugging environment for elementary programming. **International Journal of Man-Machine Studies**, *11*, 79-95.

Norman, D.A. (1986). Reflections on cognition and parallel distributed processing. In J.L. McClelland, D.E. Rumelhart, and the PDP Group (Eds.), **Parallel Distributed Processing**. 531-546.

Sack, W., Littman, D., Spohrer, J.C., Liles, A. & Soloway, E. (in prep.) **Empirical evaluation of PROUST**.

Sebrechts, M.M., LaClaire, L., Schooler, L.J., & Soloway, E. (1986). Towards Generalized Intention-based diagnosis: GIDE. In **Proceedings of the 7th National Educational Computing Conference**. San Diego, CA.

Sebrechts, M.M., Schooler, L.J., & LaClaire, L. (1986). Matching strategies for error diagnosis: A statistics tutoring aid. In **Proceedings of the International Conference on Systems, Man and Cybernetics**. Atlanta.

Shiffrin, R.M. & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. **Psychological Review**, *84*, 127-190.

Shortliffe, E.H. (1976). **Computer-based medical consultations: MYCIN**. New York: Elsevier.

Soloway, E. & Ehrlich, K. (1984). Empirical investigations of programming knowledge. **IEEE Transactions of Software Engineering**. SE-10(5).