CUCS-110-84

Interest and Predictability: Deciding What to Learn, When to Learn

Michael Lebowitz

March, 1984

This research was supported in part by the Defense Advanced Research Projects Agency under contract N00039-82-C-0427

Interest and Predictability: Deciding What to Learn, When to Learn[1]

Michael Lebowitz
Department of Computer Science
Computer Science Building, Columbia University
New York, NY 10027

Abstract

Inductive learning, which involves largely structural comparsions of examples, and explanation-based learning, a knowledge-intensive method for analyzing examples to build generalized schemas, are two major learning techniques used in AI. In this paper, we show how a combination of the two methods -- applying generalization-based techniques during the course of inductive learning -- can achieve the power of explanation-based learning without some of the computational problems that arise in domains lacking detailed explanatory rules. We show how the ideas of *predictability* and *interest* can be particulary valuable in this context.

1 Introduction

Current research in machine learning includes two relatively disparate approaches: traditional *inductive learning* from many examples (e.g., [Winston 72; Winston 80; Michalski 80; Michalski 83; Dietterich and Michalski 83; Lebowitz 83a] among many others), and a newer line of research that involves intensive application of knowledge to a single example (at a time), which we will refer to as *explanation-based learning* (e.g., [Mitchell 83; DeJong 83a; Mostow 83]; [Carbonell 83] is also related).[2] Little has been done to relate these two methods, and yet the combination seems crucial to robust learning. In this paper, we will show how two ideas, *interest* and *predictability*, can help bridge the gap. The main idea is that interest will *tell us* ***when*** to learn and predictability *what* to learn.

---

[1] This research was supported in part by the Defense Advanced Research Projects Agency under contract N00039-82-C-0427. Comments by Kathy McKeown on an earlier draft of this paper were quite useful as were discussions with Jerry DeJong.

[2] Term due to DeJong.

Much AI learning research in the past has involved *inductive learning*. While there are many varieties of inductive learning, the basic idea is that a program takes a large number of examples (or *instances*), compares them in terms of similarities and differences, and creates a generalized description based on this structural analysis. Such learning has been studied for cases where the input is specially prepared by a teacher, for unprepared input, where there are only positive instances, where there are both positive and negative instances, for a few examples, for many examples, for determining only a single concept at a time, and for determining multiple concepts. [Cohen and Feigenbaum 82] provides a good summary of this research. Pragmatically, inductive learning programs have learned by comparing examples more or less syntactically, using little "high level" knowledge of their domains (other than deciding what to represent for each instance).

In the last few years, another approach has become popular in learning research -- *explanation-based learning* (EBL). This line of research views learning as a knowledge-intensive activity, much like other tasks in AI. The basic idea is that a program takes a single example, builds up an explanation of how the various components relate to each other using traditional AI understanding or planning methods (dependent on the domain), and then loosens the properties of various components of the instance for as long as the explanation remains valid. What is left is then viewed as a generalized description of the instance that can be applied to further examples. This kind of learning is tremendously useful, as it allows generalized concepts to be determined on the basis of a single example. On the other hand, it does require extremely detailed knowledge of the domain (which may minimize the need to learn). In addition, current EBL work seems to be in the "perfect learner" paradigm that assumes that all input is typical.

The major apparent problem with EBL is that in an unrestricted domain it may be very difficult to develop the initial causal explanation that the process relies upon, particularly for systems that lack detailed domain knowledge and must use general understanding rules. Additionally, since the EBL process is computationally complex, we will not want to apply it to all examples or to all elements of examples we do consider. We suggest here a model that combines inductive and EBL methods, one that does inductive learning by noticing similarities when efficient (e.g., specific) understanding rules are not available, or when the payoff from EBL is not likely to be high, and applies EBL-type analysis at carefully selected times (on only appropriate examples and onlyappropriate parts of examples). The rules that are generalized can then be applied to understand later examples. We feel this is a promising path to robust learning, while minimizing the amount of initial domain information.

2 An EBL Example

DeJong, in [DeJong 83b], used the following story, STORY1, to illustrate explanation-based learning. We will use it to show some of the problems that can arise in doing such processing.

> **STORY1** - John, a bank teller, discovered that his boss, Fred, had embezzled $100,000. John sent Fred an inter-office memo saying that he would inform the police unless he was given $15,000. Fred paid John the money.

DeJong's program first applies standard story understanding techniques to build up a detailed causal representation of the events in STORY1. This representation includes links that show how various aspects of the *deduction* depend on each other Then DeJong repeatedly substitutes more and more general descriptions of entities in the story, for as long as the causal understanding remains valid. So, for instance, it might discover that the $15,000 could be replaced by any large amount of money and that the place of employment need not be a bank, but can be any place that involves money. The final representation with the most generalized entities that work constitutes a hypothesized "embezzlement" schema.

The EBL method works *very* well for this example, primarily because the program has a rich model of the domain, and so can build up a very detailed representation of the story Further, EBL is applied easily because the program appears to have ***only*** this relevant information. We will look at the problems that would arise if this story was embedded in a system with a wider range of information that operated on a large number of examples. In terms of the questions posed above, the problems would be deciding, from among all our examples, ***when*** to generalize, and from all the possible explanations that could be derived from a story, and all the parts of a complex explanation, deciding exactly ***what*** to generalize.

DeJong does address the "when" question. He presents five heuristics for deciding when to generalize (whether the main goal of a character is achieved, whether the goal is general, whether the resources needed by the character are generally achievable, whether the method is at least as effective as known ones, and whether the method is not already known). These heuristics are certainly related to the interest-based proposal we will make, in some sense defining "interesting" for DeJong's system. However, note that these heuristics are, like the method itself, knowledge intensive in terms of the information needed about the domain. We will consider the case where considerably less information is available for deciding when and what to generalize, or, more specifically, what instances (or generalizations made inductively), should be subjected to full EBL analysis.

Even given that a particular instance generalization should be made, we may still have a problem in deciding what aspects of the instance should be subject to generalization, and indeed how to control the process that creates the initial explanation. DeJong does not directly address this problem directly. Due to the state of his knowledge-base, he is able to simply generalize everything. Also, since he has a very complete domain model, he can make use of existing story understanding methods, as described above, to derive the initial representation. So, for example, though his system must explain why a bank executive is a plausible target for extortion, it presumably does not try to explain why all bank executives are targets of extortions (i.e., it does not start with the concept of an executive and try to explain why that requires the person to be an extortion target, since it has detailed knowledge of intentionality).

In the next section, we will show how one can deal with this problem, and then return to deciding when to generalize.

3 EBL with Less Information -- Predictability

Having outlined the EBL process, and suggested problems that might arise, we will now look at these problems in the context of the a domain typical of those that have been dealt with by inductive learning programs. UNIMEM is a program that takes a stream of facts about objects in a domain and organizes them into a generalization-based long-term memory where specific instances are stored in terms of generalized concepts [Lebowitz 80; Lebowitz 82; Schank 82; Lebowitz 83b; Lebowitz 83a]. UNIMEM learns by observation, and is not specifically provided with a set of concepts to learn. Figure 1 shows two generalized nodes in memory from a run of UNIMEM involving information about congressional districts and the voting records of their congressmen.

```
GEN5
NUC-POWER     VOTE          NO          [1]    (13)
STATE         FARM-VAL      FARS:6      [1]    (20)
NICARAGUA     VOTE          YES         [2]    (0)
HOSPITALS     VOTE          NO          [4]    (14)
DISTRICT      POP-DIR       UP          [4]    (26)
WINDFALL      VOTE          YES         [4]    (18)
STATE         MINORITY-PCT  MIN1:2      [8]    (40)
[PENNSYLVANIA22]
{ALABAMA2 CALIFORNIA14 CALIFORNIA26 CALIFORNIA34 FLORIDA6 GEORGIA1
 GEORGIA6 KENTUCKY1 KENTUCKY4 MISSISSIPPI5 NORTHCAROLINA10 PENNSYLVANIA15
 TEXAS6 TEXAS22 VIRGINIA1 VIRGINIA2 VIRGINIA4 VIRGINIA6}

    GEN6
    FOOD-STAMPS   VOTE          YES         [1]    (13)
    STATE         INCOME        INC3:4      [1]    (6)
    STATE         SEATS         GAINED      [1]    (4)
    CHRYSLER      VOTE          NO          [2]    (9)
    GAS           VOTE          NO          [2]    (9)
    SOCIAL-FUNDS  VOTE          YES         [2]    (12)
    OSHA          VOTE          YES         [2]    (14)
    CANDIDATE     PARTY         R           [2]    (8)
    PAC           VOTE          NO          [2]    (16)
    HOUSING       VOTE          NO          [3]    (13)
    [PENNSYLVANIA15]
    {CALIFORNIA14 CALIFORNIA26 CALIFORNIA20 CALIFORNIA34 FLORIDA6
     TEXAS6 VIRGINIA1 VIRGINIA4}
```

Figure 1: A Piece of UNIMEM Memory

Instances and generalizations in UNIMEM are described in terms of sets of features. The first generalization in Figure 1, GEN5, formed inductively by noticing districts with similar features, describes congressional districts with congressmen who voted "no" on bills about nuclear power and hospitals, "yes" on bills about Nicaragua and windfall profits for oil companies, where farm value is high (in the fifth of six categories; the categorization method is described in [Lebowitz 83c]), population went up between 1970 and 1980, and the minority population is relatively low. This generalization describes one district directly (the Pennsylvania 22nd) and a number of others (listed between the braces) indirectly under more specific versions of this generalization. GEN6 is one such more specific generalization, which describes moderately high income districts with Republican congressmen who voted in a particular way on several bills. GEN6 describes the Pennsylvania 16th directly, and several other districts indirectly. The numbers in parentheses in Figure 1 represent UNIMEM's confidence in a given feature (the numbers start at 0 and can rise or fall). The numbers in brackets will be explained later.

The EBL programs in the literature would approach this domain from a different angle. Presumably, they would start by looking in detail at the information from a given congressional district. The information available to UNIMEM for one such district, in the form of 32 features about the district, the state it is located in, and the votes of their congressman on a number of issues, is shown in Figure 2.

An EBL program such as DeJong's would first build up a causal analysis of the various features of the input, using whatever reasoning rules were available, and then seeing how constraints of the features could be relaxed such that the causal analysis would still hold up. So, for example, the program might decide the 22nd's congressman voted "no" on the MX issue (which was a vote to *cut* MX funding), because military spending in the district was high. Then it would see just how high the military spending had to be for the causal explanation to hold.

This approach might be appropriate if we had very thorough information about

Features: PENNSYLVANIA22 (DISTRICT)

CANDIDATE	OCCUPATION	LAW	DISTRICT	INCOME	INC2:4
DISTRICT	POP-DIR	UP	CANDIDATE	PARTY	D
STATE	IS	PENNSYLVANIA	DRAFT	VOTE	YES
NICARAGUA	VOTE	YES	MX	VOTE	NO
NUC-POWER	VOTE	NO	ALASKA	VOTE	NO
HOUSING	VOTE	YES	PAC	VOTE	NO
FOOD-STAMPS	VOTE	NO	EDUCATION	VOTE	YES
OSHA	VOTE	NO	SOCIAL-FUNDS	VOTE	NO
HOSPITALS	VOTE	NO	GAS	VOTE	YES
WINDFALL	VOTE	YES	CHRYSLER	VOTE	YES
STATE	SEATS	LOST	STATE	REGION	MA
STATE	POPULATION	POP6:7	STATE	URBAN-PCT	URB6:6
STATE	MINORITY-PCT	MIN1:2	STATE	MIGRATION	MIG1:9
STATE	SIZE	SIZ3:6	STATE	SCHOOL-EXP	SCH3:3
STATE	CRIME-RATE	CRI2:5	STATE	STATE-DEBT	DEB6:7
STATE	MILITARY-$	MIL7:9	STATE	INCOME	INC3:4
STATE	FARM-VAL	FAR5:6	STATE	TAXES	TAX2:5

Figure 2: The Pennsylvania 22d

the domain, and could construct a detailed causal explanation, particularly if there were a only limited number of points to vary in EBL's generalization phase. However, if we have only very general rules to apply, as is often the case in a new domain, then the combinatorics of the explanation process would not be feasible, particularly as it must be applied to many modified versions of the instance as constraints are relaxed.

What we propose is, first, to apply EBL methods to inductively created *generalizations*, rather than individual instances or episodes. This means that we will wait for inductive learning to suggest generalizations that will then be analyzed by EBL-type methods (i.e., a causal explanation will be derived and constraint-loosening applied). At the very least, this will avoid spending effort analyzing cases that are totally atypical. There is a larger advantage, however.

Causal explanation involves determining why a given set of conditions leads to an observed behavior. In order to do this in a domain where we have limited knowledge, we must first identify which elements of an instance are causes, and which are results. Doing so will provide a focus for applying general rules to come up with an explanation of the instance.. Most EBL systems simply posit the explanation (determine it in a fashion unrelated to the generalization phase), and needn't deal with this problem. For example, DeJong bases his EBL on a causal explanation of the sort provided by [Schank and Abelson 77; Carbonell 81; Wilensky 83], based on detailed knowledge of human intentionality. Such method's rules about human intentionality imply the causes in a situation.

If we look at generalization GEN5 in Figure 1, we see that identification of causes is not self-evident. For example, it might be that districts with high farm property values have little oil and hence their congressmen could vote safely for a windfall profits tax. Conversely, it might be that voting for a windfall profits tax actually causes the farm value to be high. (Remember, we are assuming we do not have detailed domain knowledge.)

The solution to this problem is to use ***predictability*** (presented in [Lebowitz 80; Lebowitz 83b] for indexing and understanding purposes). The basic idea is that features of a generalization that are most nearly unique to that generalization in a given context indicate the its applicability. These features are called ***predictive***. Most importantly here, the predictive features are exactly those that are likely to be causes in a causal explanation. This follows from the observation that non-predictive features occur in many generalizations, and yet are associated with many other combinations of features. Hence they cannot predict a single outcome. So, for example, if we noticed that all AI conferences were exciting, we would assume that a conference being AI causes it to be exciting, since we probably have few generalizations about AI conferences, as opposed to assuming that because an event was exciting it was probably an AI conference.[3] (See [Hempel 43; Goodman 65] for related philosophical discussion.)

If we return to Figure 1, we can see how this process might work. The numbers in brackets next to each feature indicate how many generalizations in the specified context the feature appears in. So, the "no" vote on the nuclear power issue and being a district in a state with fairly high farm value each appear only in one generalization, GEN5. In contrast, the feature "districts in states with low

[3]Note that if we knew a number of things about AI conferences, they are usually in the summer, in interesting locations, have papers in a number of areas, perhaps, then these facts would form a single conjunctive generalization.

minority percentage" appears in eight generalizations. The figures for the features in GEN6, reflect only that generalization's context, in this case the set of generalizations under GEN5.

Using the predictability information from GEN5, we can see that an explanation should be formed that shows why features like a "no" vote on nuclear power, high farm value and possibly a "yes" vote on the Nicaragua issue imply the remaining features. The other features will certainly work less well as the causes in an explanation, since they are associated with a variety of other generalizations. Similarly, an EBL program trying to explain GEN6 should look for rules that indicate why a congressman from a state with fairly high income that gained congressional seats and who voted "yes" on food stamps (the features that we assume to be the causes) should be a Republican who voted against the Chrysler guarantee, against gas controls, etc. Every single one of the predictive features may not be part of the cause, but *only* the predictive features may appear as causes. With the starting point set, we can apply general plan/goal-based understanding methods such as those in [Schank and Abelson 77; Carbonell 81; Wilensky 83], or whatever explanation methods seem appropriate.

Predictability can be applied beyond analysis of generalizations. If we did want to apply EBL to a specific instance, say the Pennsylvania 22nd district, in the manner of current EBL program, then we could use the predictability of the generalization it is stored with (GEN5, in this case) to provide starting points for the analysis. While such analysis may still be difficult, as any single instance might be anomalous in some way, at least the search will not be totally unconstrained. Note that if the instance is stored in several places in memory, which is allowable, then several possible explanations might be generated. Nonetheless, this is superior to trying to explain all combinations of cause and effect.

The point here is simply that in any given situation there are variety of different features or effects we could try to explain. Predictability provides a focus for application of general explanatory rules, even for domains with limited amounts of available world knowledge. Many problems remain in applying predictability, notably how to deal with combinations of features that are predictable even when none of the individual features are, but predictability appears to be a good place to start in building up knowledge of a domain.

4 Further Control -- Interest

Even having taken predictability into account, an EBL system will still have a large amount of work to do. We have the problem mentioned earlier of deciding when to generalize, and the explanation process could use further control. One way that people seem to deal with both of these problems is to focus on the instances that seem *interesting* to them, and the parts of the instances that are interesting. As pointed out in [Lebowitz 81], the interesting instances are exactly those that are likely to lead to successful learning.

Saying that interesting instances and interesting parts of instances are useful in learning is almost tautological. Defining interest in terms of what helps in learning would make some sense. This is the tack that [Davis 71] takes in a study of what constitutes interesting sociological theories.. But, there is another way to look at it. We will assume that interest is a *heuristic* measure of what is likely to help in terms of learning. It might well be easier to study our intuitive feel for what makes something interesting than to look directly at what is a good learning instance. We will not look formally at the components of a heuristic measure of interest in this paper (but see [Schank 79; Lebowitz 81; Lenat 82]), and instead rely on an intuitive idea of interest as we look at how interest can be applied to EBL.

If we look back at the DeJong and UNIMEM examples, we can see why interest can provide useful control. DeJong, for his extortion example, has already applied his heuristics (a form of interest), to decide that the story as a whole is interesting. Nonetheless, we can still apply the ideas of interest here. Specifically, we would want to limit the number of features in the story we actively look at when loosening constraints, or, again, the search process will become computationally unsound. So, while we certainly want to worry about the amount of money being extorted by the bank teller from his boss (it is expected to be large, but not exactly $15,000), we might not worry about the form of communication between the extorter and his victim. It is not that we would then assume the communication muse be by inter-office memo, but rather we would loosen the constraints without doing a detailed feasibility analysis. This is because our heuristics presumably show that the amount of money is interesting, while the form of communication is not.[4]

For the UNIMEM example in Figure 1, we have two potential ways to apply interest. Unlike DeJong's situation, but common to the state of many learning domains, we do not have a straightforward set of heuristics to tell us when to apply EBL. So, we will want to make use of interest. Actually, the fact that we are looking at generalizations instead of instances is one application of interest -- i.e., we are assuming that generalizations are more interesting (because they are more reliable) than individual instances. The second application of interest would be, as in the DeJong example, to decide how to focus the EBL process.

Several interest heuristics based on the ideas of relevance and novelty [Lebowitz 81] come to mind for deciding which generalizations to analyze. We would want to concentrate on generalizations that describe a number of instances, rather than just a few, and perhaps those that involve an unusual set of features. In addition, we would prefer generalizations that organize other generalizations, as they have wider applicability. So, in Figure 1, we would be more interested in GEN5 than GEN6 since it describes more instances (as well as having a number of more specific generalizations. Should it turn out that GEN6 is the only generalization involving congressmen voting "yes" on the food stamps bill and "no" on the Chrysler bailout, then it would be more interesting, because it is novel. Note that this is just what we want, since new combinations of features are likely to lead to rules we have not previously come up with.

[4]Note that since interest is idiosyncratic, what is interesting will differ from person to person, or within a learning program as it evolves over time.

Interest rules for deciding what features of a generalization on which to focus the explanation process would be similar. We would tend to focus on explaining features that are novel, but not too novel. Novel, since otherwise we can presumably just look up existing explanations, but not too novel, since we want to relate the explanations and generalizations that we derive to other parts of our knowledge base.

Note that the interest heuristics described here, as well has most of the others one can think of (at least those that do not use pre-existing domain knowledge), crucially depend on having a sizable number of instances in memory, and hence indicate the bridge between inductive learning and EBL. If processing resources are limited, interest can tell us how these resources should be expended.

If we were developing a learning system with user-imposed outside interests (e.g., "be interested in hospitals"), we can superimpose these interests on the more general heuristics in developing a system that makes generalizations relevant to the user.

To recapitulate, interest is a very intuitive idea that leads to many descriptive heuristics. If we apply these heuristics to the learning process, they will help focus on the items that accelerate learning most efficiently. We need not have a detailed understanding of why each heuristic focuses learning to make use of interest.

5 Conclusion

EBL methods hold the promise of developing learning systems that can make full use of the knowledge they already possess. However, it is necessary to relate these methods to inductive learning techniques so that our systems can not only make use of a priori knowledge, but also use similarities noticed among instances. This is particularly important in domains lacking detailed domain knowledge. We have described in this paper a three step plan involving 1) applying EBL to inductively derived generalizations, instead of individual instances, 2) using *interest* to determine when to learn, and 3) using *predictability* to help control an otherwise unmanageable explanation process. Further integration of EBL with inductive methods (not the least of which might be the application of methods for evaluating the quality of generalizations even with realistic, noisy data [Lebowitz 82; Lebowitz 83a], i.e, how we use examples *after* doing EBL), should lead to extremely powerful learning capabilities.

References

[Carbonell 81] Carbonell, J. G. *Subjective Understanding: Computer Models of Belief Systems.* UMI Research Press, Ann Arbor, Michigan, 1981.

[Carbonell 83] Carbonell, J. G. Derivational analogy in problem solving and knowledge acquisition. Proceedings of the International Machine Learning Workshop, Champaign-Urbana, Illinois, 1983, pp. 12 - 18.

[Cohen and Feigenbaum 82] Cohen, P. R. and Feigenbaum, E. A., eds. *The Handbook of Artificial Intelligence, Volume 3.* William Kaufmann, Inc, Los Altos, California, 1982.

[Davis 71] Davis, M. S. "That's interesting! Towards a phenomenology of sociology and a sociology of phenomenology." *Philosophy of the Social Sciences 1*, 1971, pp. 309 - 344.

[DeJong 83a] DeJong, G. F. An approach to learning from observation. Proceedings of the International Machine Learning Workshop, Champaign-Urbana, Illinois, 1983, pp. 171 - 176.

[DeJong 83b] DeJong, G. F. Acquiring schemata through understanding and generalizing plans. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, 1983.

[Dietterich and Michalski 83] Dietterich, T. G. and Michalski, R. S. Discovering patterns in sequences of objects. Proceedings of the International Machine Learning Workshop, Champaign-Urbana, Illinois, 1983, pp. 41 - 57.

[Goodman 65] Goodman, N. *Fact, Fiction, and Forecast.* The Boobs Merrill Company, Inc., Indianapolis, Ind., 1965.

[Hempel 43] Hempel, C. G. "A purely syntactic definition of confirmation." *Journal of Symbolic Logic 8*, 1943, pp. 122 - 143.

[Lebowitz 80] Lebowitz, M. Generalization and memory in an integrated understanding system. Technical Report 186, Yale University Department of Computer Science, 1980. PhD Thesis.

[Lebowitz 81] Lebowitz, M. Cancelled due to lack of interest. Proceedings of the Seventh International Joint Conference on Artificial Intelligence, Vancouver, Canada, 1981.

[Lebowitz 82] Lebowitz, M. "Correcting erroneous generalizations." *Cognition and Brain Theory 5*, 4, 1982, pp. 367 - 381.

[Lebowitz 83a] Lebowitz, M. Concept learning in a rich input domain. Proceedings of the International Machine Learning Workshop, Champaign-Urbana, Illinois, 1983, pp. 177 - 182.

[Lebowitz 83b] Lebowitz, M. "Generalization from natural language text." *Cognitive Science 7*, 1, 1983, pp. 1 - 40.

[Lebowitz 83c] Lebowitz, M. Classifying numeric information. Columbia University Department of Computer Science Technical Report, 1983.

[Lenat 82] Lenat, D. B. AM: Discovery in mathematics as heuristic search. In R. Davis and D. B. Lenat, Ed., *Knowledge-Based Systems in Artificial Intelligence*, McGraw-Hill, New York, 1982.

[Michalski 80] Michalski, R. S. "Pattern recognition as rule-guided inductive inference." *IEEE Transactions on Pattern Analysis and Machine Intelligence 2*, 4, 1980, pp. 349 - 361.

[Michalski 83] Michalski, R. S. "A theory of methodology of inductive learning." *Artificial Intelligence 20*, 1983, pp. 111 - 161.

[Mitchell 83] Mitchell, T. M. Learning and problem solving. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, 1983.

[Mostow 83] Mostow, J. Operationalizing advice: A problem-solving model. Proceedings of the International Machine Learning Workshop, Champaign-Urbana, Illinois, 1983, pp. 110 - 116.

[Schank 79] Schank, R. C. "Interestingness: Controlling inference." *Artificial Intelligence 12*, 3, 1979, pp. 273 - 297.

[Schank 82] Schank, R. C. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, New York, 1982.

[Schank and Abelson 77] Schank, R. C. and Abelson, R. P. *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

[Wilensky 83] Wilensky, R. *Planning and Understanding.* Addison-Wesley, Reading, MA, 1983.

[Winston 72] Winston, P. H. Learning structural descriptions from examples. In P. H. Winston, Ed., *The Psychology of Computer Vision*. McGraw-Hill, New York, 1972.

[Winston 80] Winston, P. H. "Learning and reasoning by analogy." *Communications of the ACM 23*, 1980, pp. 689 - 702.