Appears in *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pp. 664-669, Seattle, WA, July, 1994

Inductive Learning For Abductive Diagnosis*

Cynthia A. Thompson and Raymond J. Mooney
Department of Computer Sciences
University of Texas
Austin, TX 78712
cthomp@cs.utexas.edu, mooney@cs.utexas.edu

Abstract

A new inductive learning system, LAB (Learning for ABduction), is presented which acquires abductive rules from a set of training examples. The goal is to find a small knowledge base which, when used abductively, diagnoses the training examples correctly and generalizes well to unseen examples. This contrasts with past systems that inductively learn rules that are used deductively. Each training example is associated with potentially multiple categories (disorders), instead of one as with typical learning systems. LAB uses a simple hill-climbing algorithm to efficiently build a rule base for a set-covering abductive system. LAB has been experimentally evaluated and compared to other learning systems and an expert knowledge base in the domain of diagnosing brain damage due to stroke.

Introduction

Most work in symbolic concept acquisition assumes a deductive model of classification in which an example is a member of a concept if it satisfies a logical specification represented in disjunctive normal form (DNF) (Michalski and Chilausky, 1980), a decision tree (Quinlan, 1986), or a set of Horn clauses (Quinlan, 1990). However, recent research in diagnosis, plan recognition, object recognition, and other areas of AI has found that *abduction*, finding a set of assumptions that imply or explain a set of observations, is frequently a more appropriate and useful mode of reasoning (Charniak and McDermott, 1985; Levesque, 1989). This paper concerns inducing from examples a knowledge base that is suitable for abductive reasoning.

We focus on abductive diagnosis using the model of (Peng and Reggia, 1990). Given a set of cases each consisting of a list of symptoms and one or more expert-diagnosed disorders, our system, LAB (Learning for ABduction), learns a set of `disorder` $\rightarrow$ `symptom` rules suitable for abductive diagnosis, as opposed to traditional `symptoms` $\rightarrow$ `disorder` rules suitable for deductive diagnosis. Studies of human diagnosticians have demonstrated their use of abductive reasoning (Elstein et al., 1978). For example, doctors know the causes behind a patients' symptoms and when a new case is seen, they can work "backwards" given the symptoms to hypothesize the disease or diseases which are present. Abductive methods have proven useful in applications such as diagnosing brain damage due to stroke (Tuhrim et al., 1991) and identifying red-cell antibodies in blood (Josephson et al., 1987).

Abductive methods are particularly useful in domains such as these, where multiple faults or disorders are fairly common. Most inductive work on diagnosis assumes there is a single disorder (classification) for each example. One can use standard methods to learn a separate concept for each disorder that independently predicts its presence or absence; however, the effectiveness of this technique for multiple-disorder diagnosis has not been demonstrated. By finding the smallest set of disorders that globally account for all of the symptoms, abductive methods may be more appropriate for such problems.

Background on Abductive Diagnosis

Parsimonious Covering

Abduction is informally defined as finding the best explanation for a set of observations, or inferring cause from effect. A standard logical definition of an abductive explanation is a consistent set of assumptions which, together with background knowledge, entails a set of observations (Charniak and McDermott, 1985).

Our method for performing abduction is the set-covering approach presented in (Peng and Reggia, 1990). Although a simple, propositional model, it is capable of solving many real-world problems. In addition, it is no more restrictive than most inductive learning systems, which use discrete-valued feature vectors. Some definitions from their work are needed in what follows.

A *diagnostic problem* P is a four-tuple (D, M, C, M^+) where:

- D is a finite, non-empty set of objects, called disorders;

*This research was supported by the National Science Foundation under grant IRI-9102926 and the Texas Advanced Research Program under grant 003658114.

- M is a finite, non-empty set of objects, called manifestations;
- $C \subseteq D \times M$ is a causation relation, where $(d, m) \in C$ means d may cause m; and
- $M^+ \subseteq M$ is the subset of M which has been observed.

$V \subseteq D$ is called a *cover* or *diagnosis* of M^+ if for each $m \in M^+$, there is a $d \in V$ such that $(d, m) \in C$. A cover V is said to be *minimum* if its cardinality is the smallest among all covers. A cover of M^+ is said to be *minimal* if none of its proper subsets are covers; otherwise, it is *non-minimal*. The Peng and Reggia model is equivalent to logical abduction with a simple propositional domain theory composed of the rules $\{d \rightarrow m \mid (d, m) \in C\}$ (Ng, 1992). We will also write the elements of C as rules of the form $\mathtt{d} \rightarrow \mathtt{m}$. Therefore, C can be viewed as the knowledge base or domain theory for abductive diagnosis.

For the abductive portion of our algorithm, we use the BIPARTITE algorithm of (Peng and Reggia, 1990), which returns all minimal diagnoses. One immediate problem is the typically large number of diagnoses generated. Thus, following Occam's razor, we first eliminate all but the minimum covers and select one of them at random as the *system diagnosis*. The diagnosis of an experienced diagnostician is the *correct diagnosis*.

Evaluating Accuracy

We would like to have a quantitative measure of the accuracy of the system diagnosis. In the usual task, assigning an example to a single category, the accuracy is just the percentage of cases which are correctly classified. Here, we must extend this measure since each case is a positive or negative example for many disorders. Let N be the total number of disorders, C^+ the number of disorders in the correct diagnosis, and C^- the number of disorders not in the correct diagnosis, i.e., $N - C^+$. Likewise, let T^+ (True Positives) be the number of disorders in the correct diagnosis that are also in the system diagnosis, and T^- (True Negatives) be the number of disorders not in the correct diagnosis and not in the system diagnosis. *Standard accuracy* for one example when multiple diagnoses are present is defined as $(T^+ + T^-)/N$.

A second evaluation method is *intersection accuracy*. Intuitively, this is the size of the intersection between the correct and system diagnoses, as compared to the size of the diagnoses themselves. It is formally defined as $(T^+/C^+ + T^+/S)/2$, where S is the number of disorders in the system diagnosis. Third, *sensitivity* is defined by T^+/C^+, and measures accuracy over the disorders actually present, an important measure in diagnosis. Sensitivity is also called *recall* by (Swets, 1969) and others, who also define *precision* as T^+/S. Note then that intersection accuracy is the average of precision and recall. A fourth measure, *specificity*, defined as T^-/C^-, measures the accuracy over disorders not present. Sensitivity, specificity, and standard accuracy are discussed in (Kulikowski and Weiss, 1991). Finally, the accuracy of a rule base over a set of examples can be computed by averaging the appropriate score over all examples.

In a typical diagnosis, where the number of potential disorders is much greater than the number of disorders actually present ($N >> C^+$), it is possible to get very high standard accuracy, and perfect specificity, by simply assuming that all cases have no disorders. Also, it is possible to get perfect sensitivity by assuming that all cases have all disorders. Intersection accuracy is a good measure that avoids these extremes.

Problem Definition and Algorithm

The Learning for Abduction Problem

The basic idea of learning for abduction is to find a small knowledge base that, when used abductively, correctly diagnoses a set of training cases. Under the Peng and Reggia model, this may be more formally defined as follows:

Given:

- D, a finite, non-empty set of potential disorders,
- M, a finite, non-empty set of potential manifestations, and
- E, a finite set of training examples, where the ith example, E_i, consists of a set, $M_i^+ \subseteq M$, of manifestations and a set, $D_i^+ \subseteq D$, of disorders (the correct diagnosis).

Find:

The $C \subseteq D \times M$, such that the intersection accuracy of C over E is maximized.

The desire for a minimum causation relation represents the normal inductive bias of simplicity (Occam's Razor). Note we do not aim for 100% accuracy, because in some cases this is impossible, as we will discuss later. Also, we maximize intersection, not standard accuracy, for the reasons mentioned earlier.

LAB Algorithm

We conjecture that the learning for abduction problem as stated above is intractable. Therefore, we attempt to maximize accuracy by using a hill-climbing algorithm, outlined in Figure 1. Note that the rules in C always have a single manifestation rather than a conjunction of them. The first step (after initializing C) adds appropriate rules for examples with one disorder. If E_i is an example with $D_i^+ = \{d\}$ and $M_i^+ = \{m_1, \ldots, m_n\}$, then appropriate rules are $\mathtt{d} \rightarrow \mathtt{m_1}, \ldots, \mathtt{d} \rightarrow \mathtt{m_n}$. These rules must be in C if M_i^+ is to be correctly diagnosed while including a rule for each manifestation. Although in some cases this may cause incorrect diagnoses for other examples, this was not a significant problem in practice. The second step extracts all possible rules from the input examples by

Set $C = \emptyset$
For all examples with $|D_i^+| = 1$, add the appropriate rules to C
Find all potential rules, *Rules*, from E
Compute the intersection accuracy, *Acc*, of C over E
Repeat the following, until *Acc* decreases, reaches 100%, or there are no more rules:
  Initialize *bestrule* = a random $r \in$ *Rules*
  For each $R \in$ *Rules*,
    Set $C' = C \cup \{R\}$
    Compute the accuracy of C' over E
    If the accuracy of C' is greater than *Acc* then
      Set *Acc* = accuracy of C' and *bestrule* = R
  If *Acc* increased or remained the same, then
    Set $C = C \cup \{bestrule\}$
    Set *Rules* =
      $Rules - bestrule - relatedrules(bestrule)$
  Else quit and return C.

Figure 1: LAB Algorithm

adding each unique pair $\{(d, m) \mid d \in D_i^+, m \in M_i^+\}$ from each example, E_i, to *Rules*.

Next, the main loop is entered and rules are incrementally added to C until the intersection accuracy of the rule base decreases, 100% intersection accuracy is reached, or *Rules* is emptied. At each iteration of the loop, the accuracy of a rule base C' is measured. For each manifestation set, BIPARTITE is run using C' and the resulting minimum diagnoses are compared to the correct diagnosis. Note that the abduction task itself is a black box as far as LAB is concerned. Three types of accuracy are computed: intersection accuracy, standard accuracy, and sensitivity. To simulate the random selection of one minimum cover, the average accuracy of all minimum covers is determined. The best rule base is chosen by lexicographically comparing the different accuracy measures. Comparisons are first made using intersection accuracy, then standard accuracy, then sensitivity.

The remainder of the algorithm is straightforward. If all rule bases have equal accuracy, a rule is picked at random. The best rule is added to C and removed from *Rules*, along with any *related rules*. A rule, $\mathtt{d} \rightarrow \mathtt{m}$, is related to another, $\mathtt{d}' \rightarrow \mathtt{m}'$, if the two rules have the same manifestation ($m = m'$) and d and m appear only in examples in which d' and m' also appear. By removing related rules, we enforce a bias towards a minimum rule base and help maintain as high an accuracy as possible. The computational complexity of LAB can be shown to be $O(N|D|^2|M|^2)$, where N is the number of examples in E (Thompson, 1993).

Example of LAB

Let us illustrate the workings of LAB with an example. Consider the following example set, E:

E_1: D_1 ={typhoid, flu}; M_1 ={sniffles, cough, headache, fever}

E_2: D_2 ={allergy, cold}; M_2 ={aches, fever, sleepy}

E_3: D_3 ={cold}; M_3 ={aches, fever} .

First, we see that E_3 has only one disorder, so the appropriate rules are added to C, so that C = {cold→aches, cold→fever}. The intersection accuracy of this rule base is 0.583, computed as follows. For all three examples, the cover returned by BIPARTITE is {cold}. Thus, the intersection accuracy is $(0 + (1/1 + 1/2)/2 + (1/1 + 1/1)/2)/3$. Next, all possible remaining rules are formed and added to *Rules*. Then the main loop is enterered, which tests the result of adding each element of *Rules* to C. Adding the rule typhoid→sniffles to C would result in the answer {cold, typhoid} for E_1 and the answer {cold} for E_2 and E_3. Thus, the intersection accuracy of C with this rule added is 0.75. Although there are other rule bases with this same accuracy, no others surpass this accuracy, so this becomes the starting C for the second iteration. In addition, our set of *Rules* decreases, because the best rule typhoid→sniffles is removed. flu→sniffles is also removed, which is the only related rule of typhoid→sniffles. In the next iteration, the rule flu→cough, when added to C, results in the highest intersection accuracy of 0.861, because the answer for E_1 is now {typhoid, flu, cold}. So related rule typhoid→cough is also removed from *Rules*. The rule added in the next iteration is allergy→sleepy, and related rule cold→sleepy is removed. Finally, the rule typhoid→fever is added, which results in 100% intersection accuracy, and we are done. The final rule base, C, is {typhoid→fever, allergy→sleepy, flu→cough, typhoid→sniffles, cold→fever, cold→aches}. Note that no rule is associated with the manifestation headache. This is because we reached 100% accuracy before adding a rule for all symptoms, and is in keeping with our goal of learning the smallest possible rule base.

Experimental Evaluation

Method

Our hypothesis was that learning for abduction is better than learning for deduction in the case of multiple-disorder diagnosis. To test this hypothesis, we used actual patient data from the domain of diagnosing brain damage due to stroke. We used fifty of the patient cases discussed in (Tuhrim et al., 1991).[1] In this database, there are twenty-five different brain areas which can be damaged, effecting the presence of thirty-seven symptom types, each with an average of four values, for a total of 155 attribute-value pairs.

[1] We were only able to obtain fifty out of the 100 cases from the authors of the original study.

The fifty cases have an average of 8.56 manifestations and 1.96 disorders each. In addition, we obtained the accompanying abductive knowledge base generated by an expert, which consists of 648 rules.

We ran our experiments with LAB, ID3 (Quinlan, 1986), PFOIL (Mooney, to appear), and a neural network using standard backpropagation (Rumelhart et al., 1986) with one hidden layer. The neural network used has one output bit per disorder, and the number of hidden units is 10% of the number of disorders plus the number of manifestations. PFOIL is a propositional version of FOIL (Quinlan, 1990) which learns DNF rules. The primary simplification of PFOIL compared to FOIL is that it only needs to deal with fixed examples rather than the expanding tuples of FOIL.

ID3 and PFOIL are typically used for single category tasks. Therefore, an interface was built for both systems to allow them to simulate the multiple disorder diagnosis of LAB. One decision tree or DNF form is learned for each disorder. Each example $E_i \in E$ is given to the learner as a positive example if the disorder is present in D_i^+, otherwise it is given as a negative example. Thus, a forest of trees or collection of DNF forms is built.

In order to compare the performance of LAB to ID3, PFOIL, and backpropagation, learning curves were generated for the patient data. Each system was trained in batch fashion on increasingly larger fractions of a fixed training set and repeatedly tested on the same disjoint test set, in this case consisting of ten examples. At each point, the following statistics were gathered for both the training and the testing sets: standard accuracy, intersection accuracy, sensitivity, and specificity. Also, training time, testing time and concept complexity were measured.

The concept complexity of LAB is simply the number of rules in the final rule base, C. The complexity of the trees returned by ID3 is the number of leaves. This is then summed over the tree formed for each disorder. For PFOIL, the concept complexity is the sum of the lengths of each disjunct, summed again over the DNF for each disorder. Although rule, literal, and leaf counts are not directly comparable, they provide a reasonable measure of relative complexity. There is no acceptable way to compare the complexity of concepts learned by a network to these other methods, therefore no measures of concept complexity were made for backpropagation.

All of the results were averaged over 20 trials, each with a different randomly selected training and test set. The results were statistically evaluated using a two-tailed, paired t-test. For each training set size, LAB was compared to each of ID3, PFOIL, and backpropagation to determine if the differences in the various accuracy measures, train time, and and concept complexity were statistically significant ($p \leq 0.05$). If specific differences are not mentioned, they should be assumed to be statistically insignificant.

Results

Two of the resulting curves are shown in Figure 2. The left side of the figure shows the results for intersection accuracy on the testing set. LAB performs significantly better than ID3 through 15 examples, than backpropagation through 20 examples, and than PFOIL through 30 examples. Also, LAB performs significantly better than the expert knowledge base after only 15 training examples, while it takes ID3 and backpropagation 25 examples to reach this level, and PFOIL 35 examples to reach this level.

On the other hand, LAB suffers on standard accuracy for the testing set, as is seen on the right side of the figure. However, the differences between LAB and ID3 are only statistically significant for 20, 25, 35, and 40 examples. When comparing LAB to PFOIL, it is seen that PFOIL performs significantly better than LAB only at 35 and 40 examples. Also, LAB performs significantly worse than backpropagation for all training set sizes. All the systems perform significantly better than the expert knowledge base starting at 20 (or fewer) training examples.

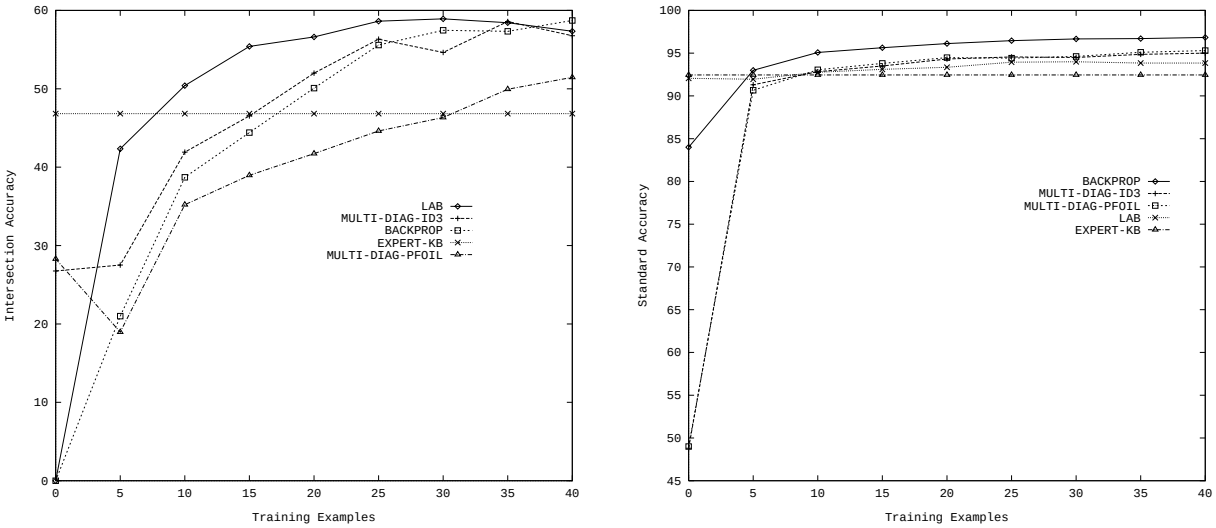
The results for sensitivity, while not shown, are also promising. LAB performs significantly better than ID3 for all training set sizes except 35, where the difference is not significant. LAB does, however, perform significantly better than PFOIL and backpropagation throughout. Also, LAB performs significantly better than the expert knowledge base starting at ten examples, ID3 does so starting at 15 examples, and PFOIL and backpropagation starting at 20 examples. For specificity, also not shown, ID3 and PFOIL perform significantly better than LAB starting at ten training examples. Backpropagation performs significantly better than LAB starting at five training examples.

Another difference in the results between the systems is in concept complexity. LAB learns a significantly more simple rule base than the trees built by ID3, but is significantly more complicated than the concepts learned by PFOIL.

Finally, for LAB the training set performance for standard accuracy starts high and stays well above 98%. On the other hand, intersection accuracy and sensitivity dip to 90%, while specificity stays above 99%. The other systems reach a training set accuracy of 100%.

Discussion

Our intuition was that obtaining a high intersection accuracy would be easier for LAB than for PFOIL or ID3. The results partially support this, in that LAB performs significantly better than all of the systems at first, then the difference becomes insignificant as the number of training examples increases. However, if a (less conservative) one-tailed, paired t-test is used instead of two-tailed, LAB's performance is significantly better than ID3 through 20 examples, and again at 30



Intersection Accuracy Standard Accuracy

Figure 2: Experimental Results on the Test Set

examples, as compared to only through 15 examples with the two-tailed test.

Also, LAB does not perform quite as well on standard accuracy compared to the other systems. However, this measure is not very meaningful, considering we get 92% accuracy just by saying that all patients have no brain damage. Finally, the sensitivity results were very encouraging, and again if we use a one-tailed, paired t-test, LAB is significantly better than ID3 for all training set sizes. Still, our results were somewhat weaker than we would have hoped. There are several possible explanations for this. First, while ID3, backpropagation, and PFOIL[2] get 100% performance on all measures on the training data, LAB does not. One possible reason is that the hill-climbing algorithm can run into local maxima.

Another reason for the difficulty in converging on the training data is that the data contain some conflicting examples from an abductive point of view. In other words, it is impossible to build an abductive rule base which will correctly diagnose all examples. One instance of these conflicts occur when there is an example, E_i, such that $|D_i^+| \geq 2$ and all $m \in M_i^+$ appear in other examples that contain only one disorder. Any attempt at an accurate abductive rule base will either hypothesize extra disorders for the examples with one disorder, or it will hypothesize a subset of the correct disorders for E_i. There are two examples with this problem in our patient data. In addition, there are other, more complicated example interactions which make it impossible to learn a completely accurate abductive rule base. This might be addressed in the future by learning more complex rules.

[2]Except at one data point.

LAB produces diagnoses during testing which include more disorders than are present in the correct diagnosis, and thus it performs well on sensitivity. On the other hand, ID3's answers include fewer disorders than the correct diagnosis, and thus performs well on specificity. These results are further indication of why ID3 performs better than LAB on standard accuracy. As mentioned previously, each example has fewer disorders than the total number possible ($D_i^+ << D$). Therefore, since ID3 is correctly predicting which disorders are *not* present more accurately than LAB, it is not surprising that it is better on standard accuracy. However, it should be emphasized that sensitivity is important in a diagnostic domain, where determining all the diseases present, and perhaps additional ones, is better than leaving some out.

Finally, we turn to concept complexity. The expert knowledge base contains 648 rules versus 111 for LAB with 40 training examples, and its performance is worse than the rules learned by LAB. There is a clear advantage, in this case, in learning rules as opposed to using expert advice. In addition, the abductive rule base is arguably easier to comprehend than either the decision tree learned by ID3 or the disjuncts returned by PFOIL, since the rules are in the causal direction. See (Thompson, 1993) for an example rule base learned by LAB.

Related Work

Since no other system learns abductive knowledge bases, no direct comparisons are possible. However, there are many systems which learn to perform diagnosis, and many abductive reasoning methods. We have already mentioned systems which learn deduc-

tive rules, both in the introduction and in our comparisons with ID3 and PFOIL. One other method that seems particularly well-suited to diagnosis is Bayesian Networks (Pearl, 1988). There have been several attempts to learn Bayesian Networks (Cooper and Herskovits, 1992; Geiger et al., 1990), but they have not been tested in realistic diagnostic domains.

Future Work

There are many opportunities for future work. First, we believe training accuracy could be improved, even given the presence of inconsistent examples. Several modifications are possible. First, different or additional heuristics could be used to improve the hill-climbing search. Second, backtracking or beam search could be used to increase training set accuracy.

A second opportunity for improvement is to reduce the number of diagnoses returned to only one during both training and testing. One way this could be done is by adding probability to abduction, as in (Peng and Reggia, 1990). Third, there is room to improve the efficiency of the system. The average training time with 40 examples is 230 seconds, versus 4 to 5 seconds for ID3 and PFOIL.

Finally, experiments in other domains are desirable; however we know of no other existing data sets for multiple-disorder diagnosis. Also, the method needs to be extended to produce more complex abductive knowledge bases that include *causal chaining* (Peng and Reggia, 1990), rules with multiple antecedents, incompatible disorders, and predicate logic (Ng, 1992).

Conclusion

Abduction is an increasingly popular approach to multiple-disorder diagnosis. However, the problem of automatically learning abductive rule bases from training examples has not previously been addressed. This paper has presented a method for inducing a set of `disorder` $\rightarrow$ `manifestation` rules that can be used abductively to diagnose a set of examples. Experiments on a real medical problem indicate that this method produces a more accurate abductive knowledge base than one assembled by domain experts, and, according to at least some important metrics, more accurate than "deductive" concepts learned by systems such as ID3, FOIL, and backpropagation.

Acknowledgments

Thanks to Dr. Stanley Tuhrim of Mount Sinai School of Medicine, and Dr. James Reggia of the University of Maryland.

References

Charniak, E. and McDermott, D. (1985). *Introduction to AI.* Reading, MA: Addison-Wesley.

Cooper, G. G. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

Elstein, A., l. Shulman, and Sprafka, S. (1978). *Medical Problem Solving - An Analysis of Clinical Reasoning.* Harvard University Press.

Geiger, D., Paz, A., and Pearl, J. (1990). Learning causal trees from dependence information. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 770–776. Boston,MA.

Josephson, J. R., Chandrasekaran, B., Smith, J. R., and Tanner, M. C. (1987). A mechanism for forming composite explanatory hypotheses. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3):445–454.

Kulikowski, C. A. and Weiss, S. M. (1991). *Computer Systems That Learn - Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems.* San Mateo, CA: Morgan Kaufmann.

Levesque, H. J. (1989). A knowledge-level account of abduction. In *Proceedings of the Eleventh International Joint conference on Artificial intelligence*, pages 1061–1067. Detroit, MI.

Michalski, R. S. and Chilausky, S. (1980). Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *Journal of Policy Analysis and Information Systems*, 4(2):126–161.

Mooney, R. J. (to appear). Encouraging experimental results on learning CNF. *Machine Learning.*

Ng, H. T. (1992). *A General Abductive System with Applications to Plan Recognition and Diagnosis.* PhD thesis, Austin, TX: University of Texas. Also appears as Artificial Intelligence Laboratory Technical Report AI 92-177.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo,CA: Morgan Kaufmann, Inc.

Peng, Y. and Reggia, J. A. (1990). *Abductive Inference Models for Diagnostic Problem-Solving.* New York: Springer-Verlag.

Quinlan, J. (1990). Learning logical definitions from relations. *Machine Learning*, 5(3):239–266.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Rumelhart, D. E., Hinton, G. E., and Williams, J. R. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing, Vol. I*, pages 318–362. Cambridge, MA: MIT Press.

Swets, J. A. (1969). Effectiveness of information retrieval methods. *American Documentation*, pages 72–89.

Thompson, C. A. (1993). *Inductive Learning for Abductive Diagnosis.* Master's thesis, Austin, TX: University of Texas at Austin.

Tuhrim, S., Reggia, J., and Goodall, S. (1991). An experimental study of criteria for hypothesis plausibility. *Journal of Experimental and Theoretical Artificial Intelligence*, 3:129–144.