INDUCTIVE LEARNING FOR ABDUCTIVE DIAGNOSIS

APPROVED:

Supervisor: _____

Raymond J. Mooney

_____

Bruce W. Porter

INDUCTIVE LEARNING FOR ABDUCTIVE DIAGNOSIS

by

CYNTHIA ANN THOMPSON, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF ARTS

THE UNIVERSITY OF TEXAS AT AUSTIN

August, 1993

Acknowledgments

I would like to express my gratitude to my advisor, Professor Raymond Mooney, for his ideas and suggestions during the course of my thesis work. Through working with him, I have gained valuable insight and experience in doing research. He provides an inspiring role model in his work and discipline.

I would also like to thank Professor Bruce Porter, for serving as my Second Reader. Third, several friends and colleagues read over my draft and offered valuable comments. Thanks to Paul Baffes, Emilio Camahort, and Bill Pierce for all their support and encouragement.

Thanks to Hwee Tou Ng for obtaining the knowledge base on brain damage and the 50 patient test cases from Dr. Stanley Tuhrim of Mount Sinai School of Medicine, New York, and Dr. James Reggia of the University of Maryland at College Park. This research was supported by the National Science Foundation under grant IRI-9102926 and the Texas Advanced Research Program under grant 003658114.

Finally, I would like to thank my family, for supporting me in all of my endevours, whatever they might be.

CYNTHIA ANN THOMPSON

The University of Texas at Austin

August, 1993

Abstract

INDUCTIVE LEARNING FOR
ABDUCTIVE DIAGNOSIS

by

CYNTHIA ANN THOMPSON, B.S.

SUPERVISING PROFESSOR: Raymond J. Mooney

A new system for learning by induction, called LAB, is presented. LAB (Learning for ABduction) learns abductive rules based on a set of training examples. Our goal is to find a small knowledge base which, when used abductively, diagnoses the training examples correctly, in addition to generalizing well to unseen examples. This is in contrast to past systems, which inductively learn rules which are used deductively. Abduction is particularly well suited to diagnosis, in which we are given a set of symptoms (manifestations) and we want our output to be a set of disorders which explain why the manifestations are present. Each training example is associated with potentially multiple categories, instead of one, which is the case with typical learning systems. Building the knowledge base requires a choice between multiple possibilities, and the number of possibilities grows exponentially with the number of training examples. One method of choosing the best knowledge base is de-

scribed and implemented. The final system is experimentally evaluated, using data from the domain of diagnosing brain damage due to stroke. It is compared to other learning systems and a knowledge base produced by an expert. The results are promising: the rule base learned is simpler than the expert knowledge base and rules learned by one of the other systems, and the accuracy of the learned rule base in predicting which areas are damaged is better than all the other systems as well as the expert knowledge base.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Intelligent activities often require one to construct explanations of phenomena. This paper views these explanatory processes as *abduction*. The philosopher C.S. Peirce (Peirce, 1958) defined *abduction* as the process of finding the best explanation for a set of observations; or inferring cause from effect. A more formal definition as it is used within artificial intelligence (AI) defines an abductive explanation as a consistent set of assumptions which, together with background knowledge, logically entails a set of observations (Charniak and McDermott, 1985). There are many situations in which abduction is used in everyday life. Physicians make inferences to conclude which diseases a patient may have based on the symptoms present and their knowledge of disease-symptom interaction. Programmers make inferences to conclude what the basic errors are in a program based on the program's behavior and their knowledge of the problem at hand. Similar types of inferences are used in other domains, such as theory formation, machine vision, legal reasoning, plan interpretation, learning, and natural language understanding. In all of these domains, the inferences made can be viewed as abduction. We present a model which will *learn* rules which will be used when reasoning abductively.

Diagnostic inference has been more well formalized and more thoroughly studied than some of these other domains. Thus, we can more easily use it as a basis for our work, and focus on the diagnostic uses of abduction. Diag-

nosis is the task of reasoning from a set of *manifestations* (findings, symptoms, observations) and potential *disorders* (causes, diseases) to a *diagnosis* (set of disorders) that explains why the manifestations are present. Diagnosis has been traditionally viewed from a deductive angle. Rules of the form `manifestations` $\rightarrow$ `disorder` are used to infer from observations which disorders are present, where "$\rightarrow$" indicates logical implication. Additional information, such as information about probability, is often attached to the implication. In real life, however, disorders *cause* manifestations; manifestations do not imply disorders. Thus, deductive rules reverse the direction of the causal link.

Instead of using the deductive angle, diagnosis can be performed from an abductive point of view. Abduction respects the direction of the causal chain and relies on a separate evaluation of potential hypotheses to choose the most plausible diagnosis. Many researchers have studied human diagnosticians and their use of abduction (Elstein et al., 1978; Kassirer, 1978; Rubin, 1975). For example, doctors know the causes behind a patients' symptoms. In other words, they can at least spell out a good approximation of the ways in which diseases cause symptoms. So when a new patient case is seen, they can work "backwards" given the symptoms to hypothesize the disease or diseases which are present. Abductive methods have proven their use in domains such as diagnosing brain damage due to stroke (Tuhrim et al., 1991) and identifying red-cell antibodies in blood (Josephson et al., 1987).

The rule bases used in diagnosis are typically large and complex. As in many other tasks, great savings are often possible by learning these rule bases instead of hand-coding them using expert knowledge. Also, the hope is that the learned rule base is more accurate than that laboriously extracted from an expert. Traditionally, research on symbolic learning for diagnosis has assumed

a deductive model of classification, in contrast to our abductive model. Rules of the form `manifestations` $\rightarrow$ `disorder` are induced from a training set, and deduction is used to diagnose future cases. For example, systems such as ID3 (Quinlan, 1986) learn decision trees (Quinlan, 1986; Breiman et al., 1984), which can be rewritten as rules of this type (Quinlan, 1987). AQ (Michalksi et al., 1986) is a second system which learns rules to be used deductively.

While inductive learning for deductive diagnosis is a well-understood problem, it has several limitations: First, much knowledge familiar to domain experts is better suited to abductive, not deductive reasoning. For example, associations between diseases and manifestations in textbooks are presented in the form "disease X may cause symptoms a,b,c...". Second, learning for deduction has trouble dealing with the case of multiple diagnosis. Most past research assumes that there is a single disorder per example, and that all disorders are disjoint. This is not the case in many real world diagnostic domains. Frequently in diagnosis, there is more than one disorder present at a time. Deduction ignores this, by looking at each disorder disjointly, to determine whether it is present. This overlooks the fact that a manifestation may have multiple causes, and that the preferred diagnosis is usually the simplest one which takes into account all manifestations.

In contrast to the systems which learn deductive rules, no previous system learns rules which are used abductively to perform classification. We have built such a system, which we believe has potential in many diagnostic application areas. This thesis presents and evaluates a method for inducing abductive rules[1] from an example set. These rules will be of the form `disorder` $\rightarrow$

[1] We will often use this term to refer to rules which are to be used in an abductive inference, and the term abductive rule (or knowledge) base to refer to a set of such rules.

`manifestation`, and will then be used abductively to diagnose unseen examples. Given a set of patient cases each consisting of a list of patient symptoms and one or more diagnosed diseases, the method attempts to learn rules which, when used abductively, correctly diagnose these cases, and generalize well to unseen cases. Our method is implemented in a system called LAB (Learning for ABduction) and has been tested on an existing data set for diagnosing brain damage due to stroke (Tuhrim et al., 1991). LAB was compared to two induction algorithms providing traditional deductive classification, one neural network, and an abductive knowledge base provided by an expert. All comparisons produced favorable results.

The remainder of this thesis is organized as follows: Chapter Two gives some background on abductive diagnosis. Chapter Three presents the problem of learning for abduction and our algorithm for solving it. Chapter Four presents experimental results and discussion. Finally, in the last three chapters, I present related work, future work, and conclusions.

Chapter 2

Background on Abductive Diagnosis

Before we go into the LAB algorithm, some background on abductive diagnosis is needed. Abduction is informally defined as finding the best explanation for a set of observations, or inferring cause from effect. Several authors (Pople, 1973; Levesque, 1989; Konolige, 1992; Ng, 1992) have proposed logical formalizations of abductive reasoning. A generic logical definition of abduction is:

Given:

A set T of axioms (the domain theory) and a conjunction O of atoms (the observations).

Find:

Minimal sets A of atoms (the assumptions) such that $A \cup T \models O$ and $A \cup T$ is consistent.

2.1 Parsimonious Covering

Our method for performing abduction on the rules learned by LAB 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, its propositional representation is not any more restrictive than most inductive learning systems, which use discrete-valued feature vectors in representing examples. 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;
- 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* of M^+ if for each $m \in M^+$, there is a $d \in V$ such that $(d, m) \in C$. Note that when we discuss covers, it will always be in the context of a diagnostic problem, so that (D, M, C, M^+) will either be stated or obvious from the context. A cover V of M^+ is said to be *minimum* if its cardinality is the smallest among all covers of M^+. A cover V of M^+ is said to be *irredundant* (*minimal*) if none of its proper subsets are also covers of M^+; it is *redundant* (*non-minimal*) otherwise. The set *causes*(m), where $m \in M$, is the set of disorders which can cause m, and is $\{d \mid (d, m) \in C\}$. The Peng and Reggia model is equivalent to logical abduction without a consistency check and with a simple propositional domain theory composed of the rules $\{d \rightarrow m \mid (d, m) \in C\}$ (Ng, 1992). (We will also often write the elements of C in the format `d` $\rightarrow$ `m`). Therefore, C can be viewed as the knowledge base or domain theory for abductive diagnosis.

For the abduction portion of our algorithm, I implemented a Lisp version of the algorithm BIPARTITE given by (Peng and Reggia, 1990). The pseudocode for BIPARTITE is given in Figure 2.1. The input is a set of manifestations, M^+, and a causation relation, C. The output is a compact representa-

```
begin BIPARTITE
      hypothesis := {∅}
      For each m ∈ M+ do
          hypothesis := revise(hypothesis, causes(m))
      return hypothesis
end BIPARTITE

revise(G, H1)
      F := div(G, H1)
      Q := augres(G, H1)
      return F ∪ res(Q, F)
```

Figure 2.1: BIPARTITE Algorithm

tion of "the set of all irredundant [minimal] covers of all manifestations known to be present" (Peng and Reggia, 1990). BIPARTITE works in an incremental fashion, looking at one manifestation at a time, and incorporating its causes into the diagnosis so far. The functions `div`, `res` and `augres` perform the following functions. `Div`, which is compared to a set division operation by Peng and Reggia, gets all minimal covers which cover *both* O and $O \cup \{m\}$, where O is the old manifestation set, and m is the newest manifestation encountered. Note that covers for only $O \cup \{m\}$ may be missed by this operation. The `augres` operation is described as an augmented residual of division. The `augres` result, Q, contains all minimal covers of $O \cup \{m\}$ which are non-minimal covers of O. However, Q might also contain some non-minimal covers of $O \cup \{m\}$. Therefore, the `res` operation (described as residual of division) removes these redundant covers. By combining the results of `div` and `res`, all covers of $O \cup \{m\}$ can be constructed from all covers of O and from the disorders in $causes(m)$. See (Peng and Reggia, 1990) for details on BIPARTITE.

m	*hypotheses*
sniffles	(cold $\vee$ pneumonia $\vee$ allergy $\vee$ hay-fever)
fever	(cold $\vee$ pneumonia $\vee$ (hay-fever $\wedge$ typhoid) $\vee$ (allergy $\wedge$ typhoid))
cough	((cold $\wedge$ lung-cancer) $\vee$ (pneumonia $\wedge$ lung-cancer) $\vee$ (cold $\wedge$ emphysema) $\vee$ (pneumonia $\wedge$ emphysema) $\vee$ (cold $\wedge$ typhoid) $\vee$ (pneumonia $\wedge$ typhoid) $\vee$ (allergy $\wedge$ typhoid) $\vee$ (hay-fever $\wedge$ typhoid))
headache	((allergy $\wedge$ typhoid) $\vee$ (pneumonia $\wedge$ emphysema) $\vee$ (pneumonia $\wedge$ typhoid) $\vee$ (pneumonia $\wedge$ lung-cancer) $\vee$ (hay-fever $\wedge$ typhoid $\wedge$ brain-tumor) $\vee$ (hay-fever $\wedge$ typhoid $\wedge$ flu) $\vee$ (cold $\wedge$ emphysema $\wedge$ flu) $\vee$ (cold $\wedge$ emphysema $\wedge$ brain-tumor) $\vee$ (cold $\wedge$ typhoid $\wedge$ flu) $\vee$ (cold $\wedge$ typhoid $\wedge$ brain-tumor) $\vee$ (cold $\wedge$ lung-cancer $\wedge$ flu) $\vee$ (cold $\wedge$ lung-cancer $\wedge$ brain-tumor) $\vee$ (cold $\wedge$ emphysema $\wedge$ allergy) $\vee$ (cold $\wedge$ lung-cancer $\wedge$ allergy))

Table 2.1: BIPARTITE trace

2.2 Example of BIPARTITE

Let us give an example of the workings of BIPARTITE. Although we are illustrating BIPARTITE, this is also an example of abduction in general. Say that we have the following rules in C:

`cold`$\rightarrow$`sniffles, pneumonia`$\rightarrow$`sniffles, allergy`$\rightarrow$`sniffles,`

`hay-fever`$\rightarrow$`sniffles,`

`cold`$\rightarrow$`fever, pneumonia`$\rightarrow$`fever, typhoid`$\rightarrow$`fever,`

`emphysema`$\rightarrow$`cough, typhoid`$\rightarrow$`cough, lung-cancer`$\rightarrow$`cough,`

`pneumonia`$\rightarrow$`headache, allergy`$\rightarrow$`headache, flu`$\rightarrow$`headache,`

`brain-tumor`$\rightarrow$`headache.`

Also, let $M^+ =$ { `sniffles, fever, cough, headache` }. Table 2.1 shows the output of each iteration through the loop, where the manifestations are shown in the left column. We follow the standard notation of $\vee$ for `or` and $\wedge$

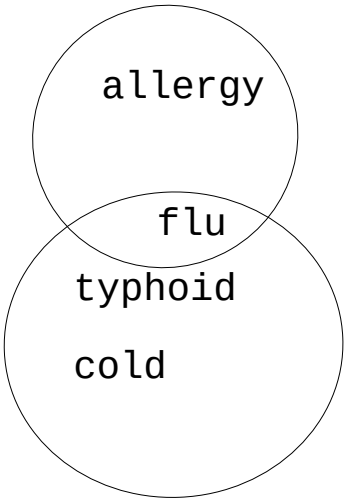
for `and`. The first iteration returns the hypothesis that either `cold`, `pneumonia`, `allergy`, or `hay-fever` is present. There are a total of fourteen distinct diagnoses after the last iteration.

Thus, the answer returned by BIPARTITE represents all minimal diagnoses (covers). These covers could now be analyzed by a diagnostician. One immediate problem is the large number of explanations generated. As we can see from the example above, there are typically a large number of minimal covers of a manifestation set. Thus, following Occam's razor, we might first eliminate all but the smallest explanations (the minimum covers). In the example above, there are only four minimum covers, compared to fourteen minimal covers. These are: `(allergy` $\wedge$ `typhoid)` $\vee$ `(pneumonia` $\wedge$ `emphysema)` $\vee$ `(pneumonia` $\wedge$ `typhoid)` $\vee$ `(pneumonia` $\wedge$ `lung-cancer)`. Eliminating all but the minimum covers is one step we decided to take in what follows. This is mainly a heuristic measure, meant to keep the number of diagnoses as low as possible. Below, when we discuss covers, answers of BIPARTITE, or diagnoses, we are referring to this set of minimum covers, unless otherwise specified. Then, to determine which of the minimum covers is the diagnosis for the patient, a naive diagnostician could only guess which to use. In what follows, we will view this as a randomly chosen diagnosis and call it the *system diagnosis*. For example, we could choose `(pneumonia` $\wedge$ `emphysema)` as our system diagnosis from the four possibilities given above. If we wanted to evaluate the system diagnosis, we could then compare it to the diagnosis of an experienced diagnostician, which we will call the *correct diagnosis*.

2.3 Evaluating Accuracy

We would like to have some numerical measure of the accuracy of the system diagnosis. One evaluation method is *standard accuracy*. In standard classification tasks, where there are only two classes, + or −, indicating whether the example is a member of the class of interest, it is a simple matter to compute accuracy: it is just the percentage of cases which are correctly classified. Therefore, each example is either totally correct or totally incorrect. Here, we must extend this measure for more than two classes, since each case is a positive or negative example for many disorders. In the following, C^+ will stand for the number of disorders in the correct diagnosis, and C^- will stand for the total number of disorders possible minus C^+, e.g., the number of disorders which aren't in the correct diagnosis. Likewise, T^+ (True Positives) will stand for the number of disorders in the correct diagnosis which are also in the system diagnosis, and T^- (True Negatives) for the number of disorders not in the correct diagnosis which are also not in the system diagnosis. With this in mind, standard accuracy for one example, when multiple diagnoses are present is the percentage of the total set of disorders which are correctly predicted as either present or absent. It is computed by the formula $(T^+ + T^-)/(C^+ + C^-)$. Standard accuracy is illustrated in Figure 2.2, along with the other accuracy measures described below. The top circle contains the correct diagnosis: `allergy` and `flu`, and the bottom circle contains the system diagnosis: `flu`, `typhoid`, and `cold`. In this example, we are assuming there are a 25 different disorders which could be present, so $(C^+ + C^-) = 25$, and $T^- = 23 - 2 = 21$.

A second evaluation method is *intersection accuracy*. Intersection accuracy is the percentage of disorders in the correct diagnosis which are correctly



Correct Diagnosis
= allergy, flu

R = 2

System Diagnosis
= typhoid, cold, flu

S = 3

I = 1

Intersection Accuracy =
$\frac{1/2 + 1/3}{2}$ = .417

Standard Accuracy=
$\frac{1 + 21}{25}$ = .88

Sensitivity =
1/2 = .5

Specificity =
21/23= .913

Figure 2.2: Accuracy Measures Example

predicted, averaged with the percentage of disorders in the system diagnosis which are correctly predicted. It is computed by the formula $(I/R + I/S)/2$, where R is the number of disorders in the correct diagnosis, S is the number of disorders in the system diagnosis, and I is the size of the intersection of the correct diagnosis and the system diagnosis.

Third, we can measure the *sensitivity* of an answer by computing the value of the formula T^+/C^+. Sensitivity measures accuracy over the disorders actually present, an important measure in diagnosis. Sensitivity is intuitively important in the brain damage domain, and indeed in other diagnostic domains. This is because saying someone is well when they are actually sick can be harmful to their health, while the opposite mistake, saying they are sick when they are well, is a less serious error.

A fourth measure, *specificity*, defined as T^-/C^-, measures the accuracy over disorders not present, or how successful we are at saying someone is well when they are well. In Figure 2.2, we see that specificity is higher than

standard accuracy, since there are 23 (out of 25 total) diseases not present in the correct diagnosis, and we correctly predicted 21 of them as missing. Most of these measurements (except intersection accuracy) are discussed in (Kulikowski and Weiss, 1991). Finally, when the system diagnosis is derived by running BIPARTITE with underlying rule base C, we can also define each of these measures with respect to a rule base C over an example set E, by taking the average of the measure over the number of examples.

In a typical diagnosis, where the number of possible disorders is much greater than the number of disorders actually present in a case, it is possible to get very high standard accuracy, and perfect specificity, just by saying that all cases have no disorders. Also, it is possible to get perfect sensitivity by saying that all cases have all disorders. Intersection accuracy is a good measure that avoids these extremes.

Chapter 3

Problem Definition and Algorithm

3.1 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.

Find:

The smallest causation relation, $C \subseteq D \times M$, 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 in Secion 4.3.

3.2 LAB Algorithm

We hypothesize that the learning for abduction problem, as it is stated above, is intractable. Therefore, we attempt to reach the maximum accuracy by using a hill-climbing algorithm, LAB. The LAB algorithm is outlined in Figure 3.1. We will call the elements $(d, m) \in C$ *rules*. The algorithm attempts to maximize, at each step, the accuracy of the current rule base. The first step (after initializing C) adds to C 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 for E_i are $(d, m_1), \ldots, (d, m_n)$. We know these rules must be in C, because they are all needed to correctly diagnose M_i while including a rule for all manifestations. The second step extracts all possible rules from the input examples by simply adding every possible pair $\{(d, m) \mid d \in D_i^+, m \in M_i^+\}$ from each example, E_i, to *Rules*, without repetitions.

Next, the main loop is entered in which rules are added to C until the intersection accuracy of the rule base is decreased by adding further rules, until we have reached 100% intersection accuracy, or until *Rules* is empty. At each iteration of the loop, the accuracy of a rule base C' is measured in the following manner: For each manifestation set, the BIPARTITE algorithm is run using C' as the rule base. (Note that, although we use BIPARTITE as our abduction mechanism, the abduction task itself is a black box as far as LAB is concerned.) The minimum diagnoses returned are compared to the correct diagnosis. Three types of accuracy are computed at this stage: intersection accuracy, standard accuracy, and sensitivity.

As we discussed earlier, intersection accuracy is a good measure to attempt to maximize in our domain. Therefore, we sort the rule base, using the different accuracy measures to do a lexical sort. Comparisons are first made

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 accuracy, *Acc*, of C over E
Repeat the following, until *Acc* decreases, reaches 100%, or there are no more rules:
    Initialize *rule-to-pick* = 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* to accuracy of C'
            set *rule-to-pick* to R.
    If *Acc* increased or remained the same, then
        set $C = C \cup \{rule\text{-}to\text{-}pick\}$
        set $Rules = Rules - \{rule\text{-}to\text{-}pick\}$
            $-\{related\text{-}rules(rule\text{-}to\text{-}pick)\}$
    else quit and return C.

Figure 3.1: LAB Algorithm

between the intersection accuracy of the best rule base so far and C', then, if these are the same, the comparison is made for standard accuracy, then sensitivity. To simulate the random selection of one minimum cover, we take an average of each accuracy measure over all covers returned by BIPARTITE. See Section 2.3 for a further discussion of the computation of these measures.

The remainder of the algorithm is straightforward. If all rule bases have equal accuracy, a rule is picked randomly. The best rule is added to C and removed from *Rules*, along with any *related rules*. A rule, (d, m), is related to another, (d', m'), if the two rules have the same manifestation ($m = m'$) and d and m do not appear together in any examples other than those in which d' and m' appear. By removing from *Rules* the related rules of the best rule, we

enforce a bias towards a minimum rule base, and help keep the accuracy as high as possible. If these rules were kept, and we calculated the accuracy of adding them to C, the accuracy would remain the same as before. This is because they would generate an additional diagnosis, containing d, for examples in which m occurs, besides the diagnosis we already had in which d' appears. Thus, we have two (or more) equally correct diagnoses for these examples. If no other rule increases the accuracy (but many keep the accuracy the same), we would be likely to choose one or more of these related rules to add to C, since we only quit adding rules when the accuracy decreases. Then it would be harder to find rules later which would increase the accuracy again.

Finally, the process of adding rules continues until no rules are left, the accuracy is 100%, or no rule can improve the accuracy of C.

3.3 Example of LAB

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

E_1: Disorders: `typhoid, flu`; Manifestations: `sniffles, cough, headache, fever`

E_2: Disorders: `allergy, cold`; Manifestations: `aches, fever, sleepy`

E_3: Disorders: `cold`; Manifestations: `aches, fever`.

First, we see that E_3 has only one disorder, so we add the appropriate rules to C, so that C = `(cold→aches, cold→fever)`. The intersection accuracy of this rule base is .583. This is computed as follows. For all three examples, the cover returned by BIPARTITE is `(cold)`. Thus, the intersection accuracy

of C is $(0 + (1/1 + 1/2)/2 + (1/1 + 1/1)/2)/3 = 0.583$. Next, all possible remaining rules are formed and added to *Rules*. Then the main loop is entered, which tests the result of adding each element of *Rules* to C. For example, the rule base C' = (`typhoid`→`sniffles, cold`→`aches, cold`→`fever`) 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' is $((1/2 + 1/2)/2 + (1/1 + 1/2)/2 + (1/1 + 1/1)/2)/3 = 0.75$. Although there are other rule bases with this same accuracy, no others surpass this accuracy, so C' becomes the starting C for the second iteration through the loop. In addition, our set of *Rules* decreases, because we remove `flu`→`sniffles`, which is the only related rule of `typhoid`→`sniffles`. This time through the loop, we see that the rule `flu`→`cough`, when added to C, results in the highest intersection accuracy, of .861, because the answer for E_1 is now (`typhoid flu cold`), so we get intersection accuracy of $((2/3+2/2)/2+(1/1+1/2)/2+(1/1+1/1)/2)/3 = 0.861$, and related rule `typhoid`→`cough` is removed from *Rules*. The rule added the next time through the loop is `allergy`→`sleepy`, causing the accuracy to increase to .944, and related rule `cold`→`sleepy` is removed. Finally, we add the rule `typhoid`→`fever`, 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 learning a rule for all symptoms present, and is in keeping with our goal of learning the smallest C possible.

3.4 Computational Complexity Evaluation

Let us examine the computational complexity of our problem. Let $|D|$ be the size of our disorder set and let $|M|$ be the size of our manifestation set. Then the size of the search space of possible rule bases is $O(2^{|D||M|})$. This is derived from the fact that if all disorder-manifestation pairs were actually present somewhere in the examples, then every such pair would be a potential rule to add to C. If we were to make a blind breadth-first search to accomplish our task, we would have to look at all rule bases of size one, find their accuracy, then do the same for all of size two, then three, etc., until all rule bases have been evaluated. Since this problem grows exponentially with the example set, it is intractable in general. Note, though, that we don't actually need to look at all rules in $D \times M$, but only a plausible subset, where each (d, m) examined is in E_i for each $E_i \in E$. However, this can still grow exponentially with E.

This approximation of exponential growth could be an over-estimate, because we do not need to perform a blind search. Conversely, although we have no proof, we hypothesize that our problem is NP-Hard because of the difficulty of finding a minimum rule base such that the accuracy is maximized. Therefore, it is useful to examine the computational complexity of LAB. The first significant step, finding all *Rules*, takes $O(|D||M|)$ time. Next, the number of times that BIPARTITE is executed is $O(N|D|^2|M|^2)$, where N is the number of examples in E. This upper bound is derived from the summation

$$\sum_{r=0}^{|D||M|-1} N(|D||M| - r),$$

where r is the number of rules added to the rule base so far, and $(|D||M| - r)$ is the number of rules left from which to choose. Since we run BIPARTITE once for each example, for each rule base C', we multiply the number of rules left

from which to choose by N. Finally, in the worst case, we could add all possible rules in $D \times M$.

Now we have the number of times which BIPARTITE is run, but we have not discussed the complexity of BIPARTITE. Finding all *minimum* covers of a given M_i^+ is NP-hard (Peng and Reggia, 1990). The complexity of BIPARTITE itself is $O(2^{M_i^+})$ for a particular example E_i, since we are finding all minimal covers. However, in most patient cases in our domain (and presumably in most diagnostic domains), the number of manifestations is relatively small (less than ten). Therefore, it is actually the repeated execution of BIPARTITE that dominates our time, not the BIPARTITE algorithm itself. The only other computations in the loop are those of the accuracy, which are included in the analysis already, since they are performed as often as bipartite is run.

Chapter 4

Experimental Evaluation

4.1 Method

Our aim is to show that learning for abduction is better than learning for deduction in the domain of multiple-disorder diagnosis. To test our hypothesis, we used actual patient data from the domain of diagnosing brain damage due to stroke. We used fifty[1] of the patient cases discussed in (Tuhrim et al., 1991). In this database, there are twenty-five different brain areas which can be damaged (e.g., right midbrain), effecting the presence of thirty-seven symptom types (e.g., gait type), each with an average of four values, for a total of 155 total attribute-value pairs possible (e.g., gait type = unsteady). These fifty cases were used as the example set, E, and 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 (hereafter called the expert KB, or original KB).

We ran our experiments with LAB, ID3, PFOIL, and a neural network. ID3 (Quinlan, 1986) learns a decision tree for classifying examples into multiple categories, and uses an information gain criterion for building the tree. In order to make it comparable to LAB and PFOIL, no pruning is performed. PFOIL

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

(Mooney, to appear) is a propositional version of FOIL (Quinlan, 1990). FOIL is a system for learning first-order Horn clauses; however the basic algorithm is a heuristic covering algorithm for learning DNF expressions. 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. The neural network was trained using standard backpropagation (Rumelhart et al., 1986) with one hidden layer.

ID3 and PFOIL are typically used for single class tasks, where each example represents only one category. In other words, there may be multiple categories to choose from, but each example can only belong to one of the multiple categories. In our domain, each example may belong to multiple categories (or disorders). Therefore, an interface was built for both systems to allow them to simulate the multiple disorder diagnosis of LAB. One tree or DNF form for one disorder is learned at a time. For each disorder, an example E_i 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. This is done for all examples in E. 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, specificity, and the number of diagnoses returned. 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 complexity were statistically significant ($p \leq 0.05$).

4.2 Results

The resulting curves are shown in Figures 4.1-4.5. All run times are for a SPARCstation 2 running Lucid Common Lisp. When given zero training examples, LAB and PFOIL classify everything as negative, and ID3 returns a random class. Differences at zero training examples are therefore not particularly meaningful and will not be considered in the following discussion. If specific differences are not mentioned, they should be assumed to be statistically insignificant.

The top of Figure 4.1 shows the results for intersection accuracy on the testing set. LAB performs significantly better than ID3 through fifteen examples, than backpropagation through 20 examples, and than PFOIL through

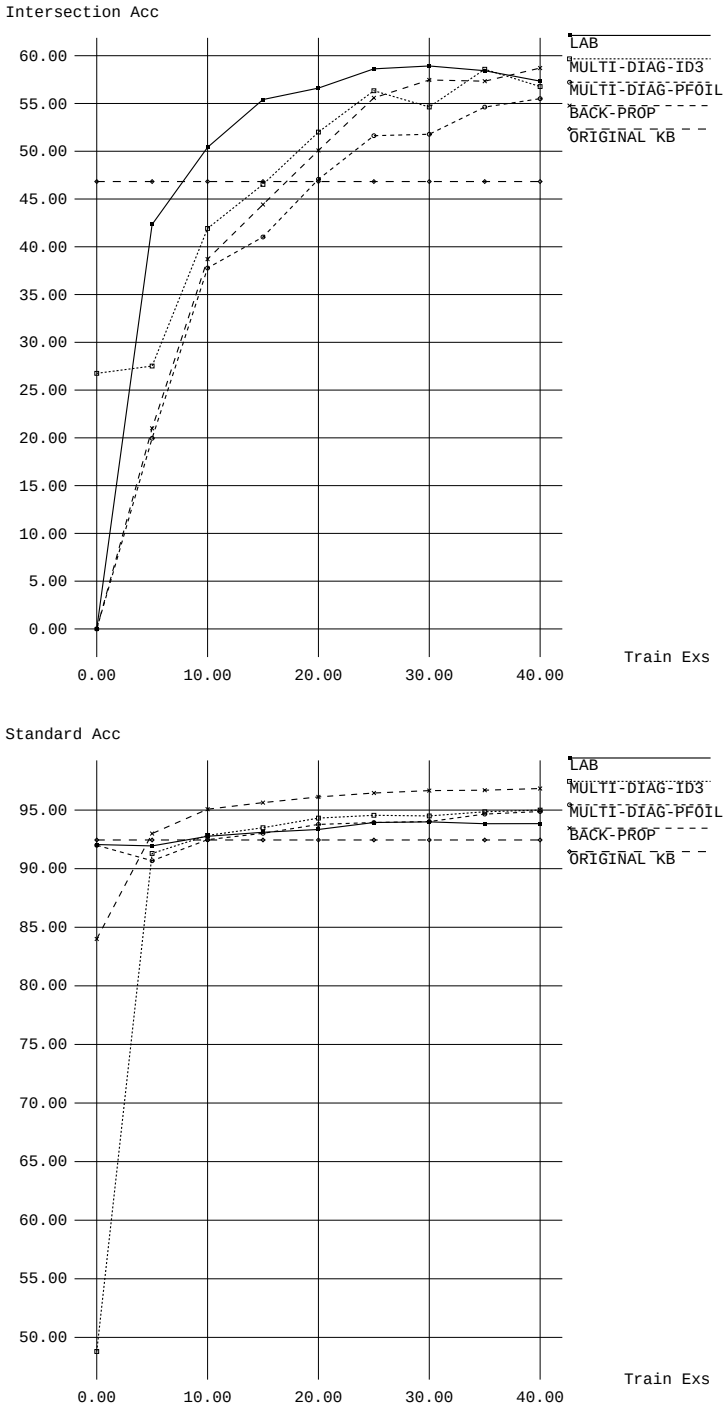


Figure 4.1: Intersection Accuracy and Standard Accuracy

30 examples. Also, LAB performs significantly better than the expert KB after only fifteen training examples, while it takes ID3 and backpropagation twenty-five examples to reach this level, and PFOIL takes 35 examples to reach this level.

On the other hand, LAB suffers on standard accuracy for the testing set, as we can see on the bottom half of the figure. However, the differences between LAB and ID3 are only statistically significant for 20, 25, 35, and 40 examples. When we compare LAB and PFOIL, we can see 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 KB starting at twenty (or fewer) training examples. In addition, all systems perform significantly better than the 92.16% standard accuracy possible in this domain by guessing all patients are well, starting at 15 (or fewer) examples.

The training performance for standard accuracy is shown in Figure 4.2. While the downward trend looks alarming, note that we do stay well above 98% standard accuracy. On the other hand, intersection accuracy and sensitivity (not shown) dip to 90%, while specificity (not shown) stays above 99%.

The results for sensitivity and specificity are shown in Figure 4.3. For the sensitivity measure, as shown on the top of the figure, LAB performs significantly better than ID3 for all example 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 KB starting at ten examples, ID3 does so starting at 15 examples, and PFOIL and backpropagation starting at 20 examples. For specificity, on

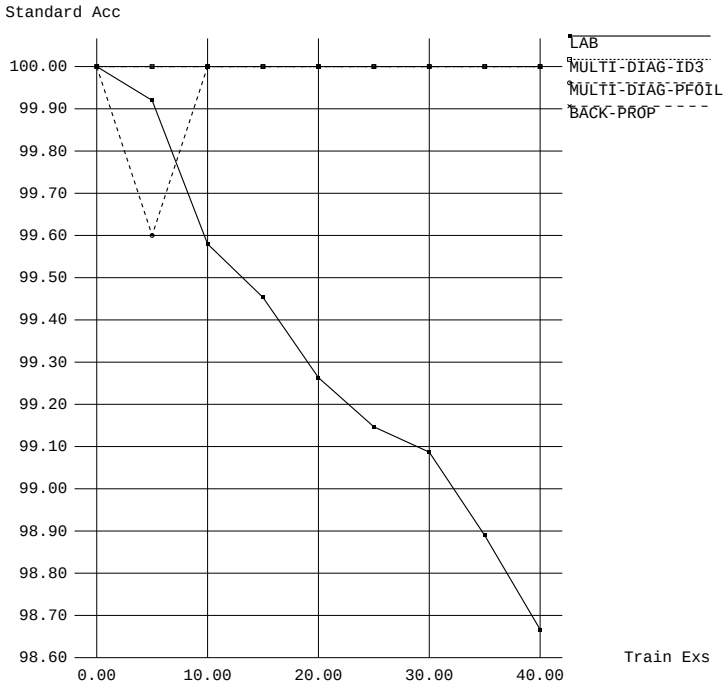


Figure 4.2: Training Accuracy

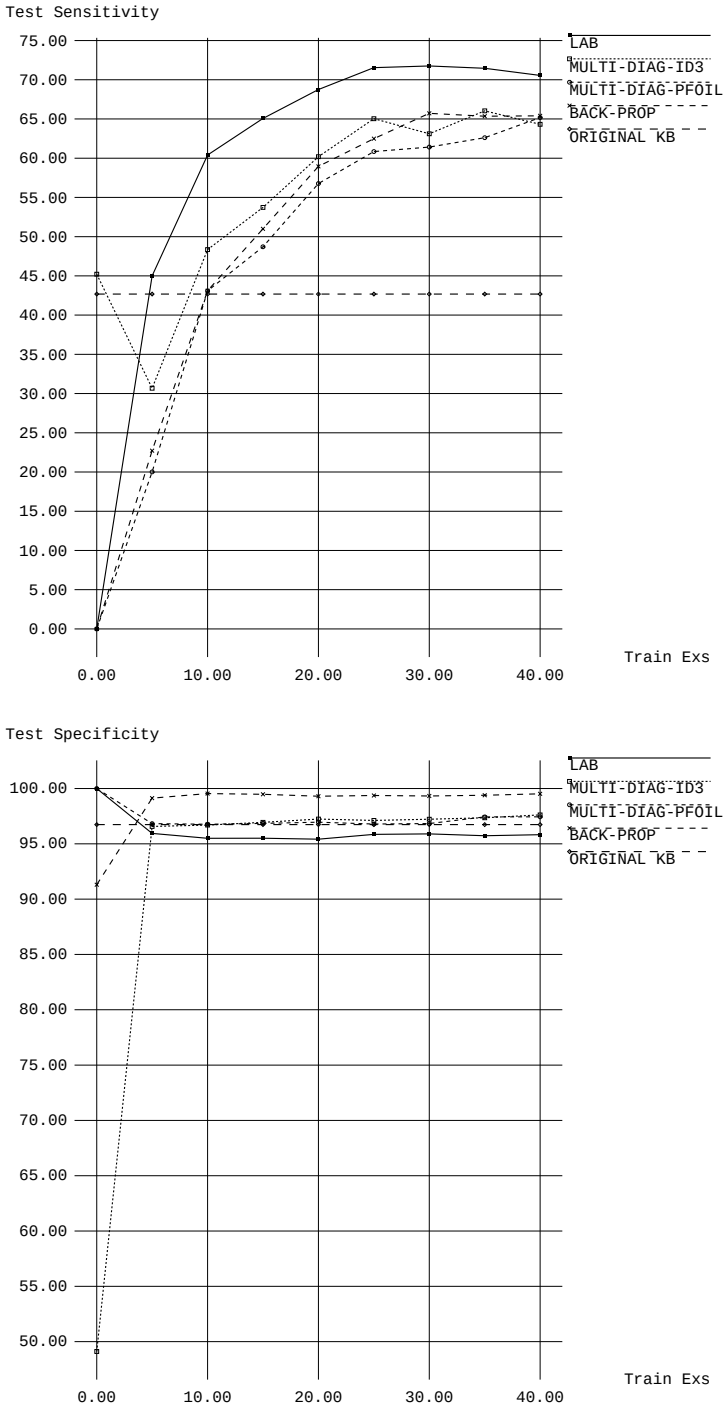


Figure 4.3: Testing Sensitivity and Specificity

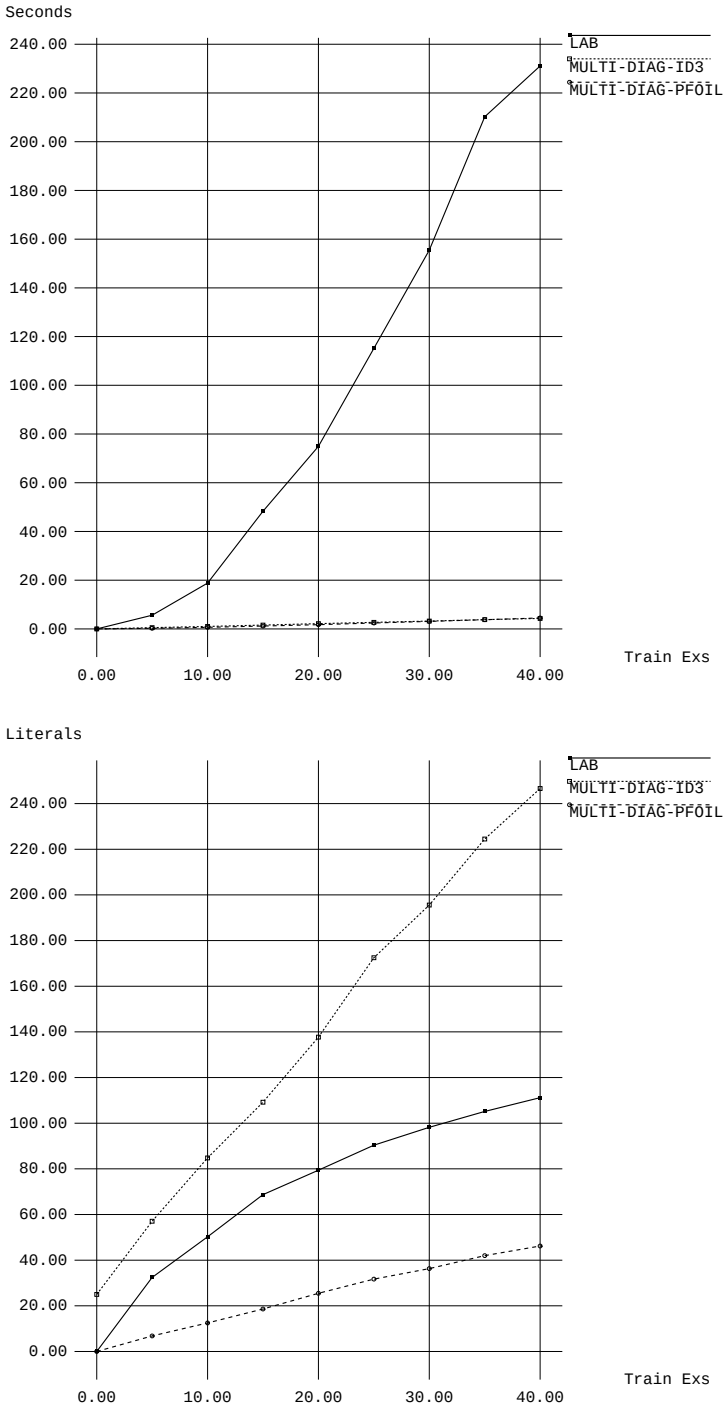


Figure 4.4: Train Time and Concept Complexity

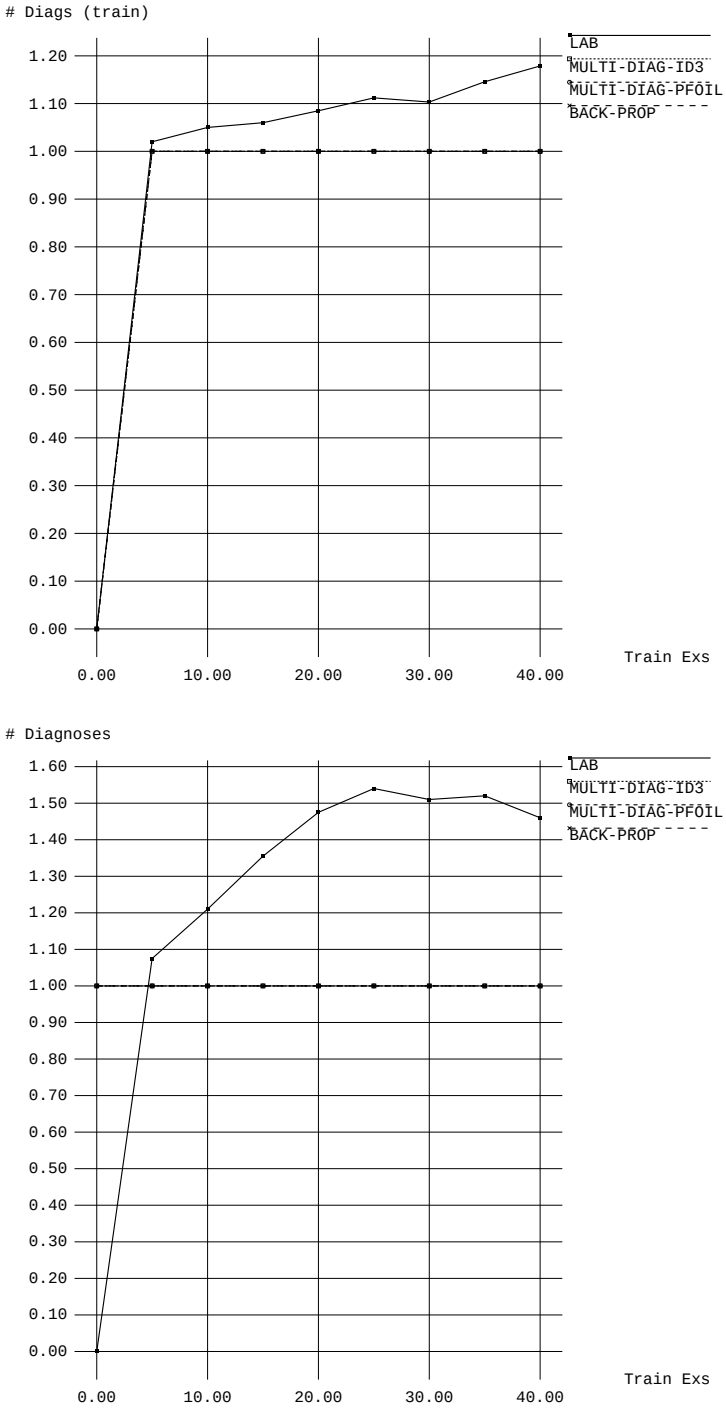


Figure 4.5: Number of Diagnoses Returned

the bottom, ID3 and PFOIL perform significantly better than LAB starting at ten training examples. Backpropagation performs significantly better than LAB starting at five training examples. LAB never surpasses the specificity performance of the expert KB, but backpropagation does, significantly, starting at five examples. ID3 performs significantly better than the expert KB after 30 examples, and PFOIL performs significantly better at forty examples only.

Another difference in the results between the systems can be seen in the training time and concept complexity. Figure 4.4 displays these differences. The training time for backpropagation is not shown, but went as high as 1751 seconds for 40 training examples. The training time for LAB (on the top of the figure) was significantly greater than that for either ID3 or PFOIL. However, as we can see, the time at first looks to be growing polynomially, but later looks to be linear. One reason for this is that we are reaching a saturation point on the number of potential rules which need to be examined.

The bottom of the figure displays the more positive results for 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, we want to look at the number of diagnoses returned by LAB during training and testing. This can be seen in Figure 4.5. The number of diagnoses returned per example is reasonably close to one during training (shown at the top of the figure), and closer to 1.5 during testing.

4.3 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 we use a one-tailed paired t-test instead of two-tailed, LAB's performance is significantly better than ID3 through 20 examples, and again at 30 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. Several explanations for this can be found. 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 we are using a hill-climbing algorithm, and can run into local maxima.

Another reason for the difficulty during training is that the patient cases 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 the manifestations of D_i appear in other examples which contain only one disorder. Following is an example from

[2]Except at one data point.

our patient data. See the appendix for the meanings of the abbreviations.

E_i: Disorders: `lfl, lic`; Manifestations: `at(hr), fst(rc), ds(mod), wt(hr), as(ri)`;

E_j: Disorders: `lfl`; Manifestations: `fst(rc), ts(r), ds(mod), wt(hr), as(ri)`;

E_l: Disorders: `lfl`; Manifestations: `ded(d), at(hr), fst(rc), wt(hr), des(rmi), as(ri), bs(r)`

Any abductively used rule base we build will either hypothesize extra disorders for E_j or E_l, or it will hypothesize a subset of the correct disorders for E_i. Which one happens depends on the order in which we test rules for addition to C. There are two examples with the qualities of E_i in our patient data. In addition to this type of example, there are other, more complicated example interactions which make it impossible to learn a totally accurate abductive rule base.

So, another reason for the dip in training accuracy is that the more examples there are in the training set, the more likely that conflicting examples will occur in the data. This makes it harder to achieve our global goal, since the algorithm does not know the difference between conflicting examples and those that it can learn correctly. We ran another test on just 40 examples in which we did not see any conflicts. This time, even with the hill-climbing algorithm, 99% standard accuracy is maintained on the training data after 30 examples. Also, intersection accuracy stays above 91%, and sensitivity above 96%. However, testing performance suffers in comparison to the original study, and thus the results were analyzed no further.

In addition to the fairly good results on intersection accuracy, there are other positive results. While ID3 and PFOIL get 100% intersection accuracy and sensitivity on the training set and LAB does not, LAB performs better than both systems on these measures during testing. We have already discussed the intersection accuracy results, and now turn to the results for sensitivity. Recall first that sensitivity measures accuracy over the disorders actually present in a case, while specificity measures accuracy over disorders not present.

LAB produces diagnoses during testing that are overly general, and thus performs well on sensitivity, while ID3 and PFOIL's answers are overly specific, and thus they perform well on specificity. These results are further indication of why ID3 performs better than LAB on standard accuracy. Each example contains few disorders compared to the number possible. 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. One reason for this is that as more training examples become available, ID3 uses more information about the absence of manifestations to make its decisions as to whether a disorder is present. In some cases, ID3 hypothesizes that if a certain manifestation is present, a certain disorder is not present, no matter what other manifestations are present. This causes ID3 to miss the fact that the manifestation could be present due to a different disorder. LAB can not use information about the absence of a manifestation to make predictions like this. LAB weighs all manifestations more equally and thus has a tendency to hypothesize a disorder present because of one manifestation when ID3 would hypothesize that the presence of another manifestation indicates it is not present. Therefore, LAB is typically predicting at least as many disorders as are actually present.

Since abduction must explain all manifestations present for which it knows a rule, it is always forced to hypothesize at least one disorder per case, as long as at least one manifestation present is in the rule base. ID3 has no such bias. It evaluates the presence of each disorder independently of the others. BIPARTITE increases or decreases the number of disorders present as it examines each manifestation, but this number can never go below one.

We see some additional reasons for LAB to have an overly general bias: Suppose we had a rule base, C_1 which correctly diagnosed half of our examples, but which for the other half returned a diagnosis which was correct, but in addition had an incorrect disorder present. Suppose also there was a second rule base, C_2 which correctly diagnosed half of our examples, but which for the other half returned two diagnoses, one correct, and one with a disorder missing and a different one in its place. A better intersection accuracy is obtained for C_1 than for C_2 in our example set. This indicates that most examples will contain more disorders than are needed, instead of more diagnoses than are needed. This is another reason why specificity is low for LAB but sensitivity is higher.

With respect to concept complexity, as the expert KB contains 648 rules (versus 111 for LAB with 40 training examples) and performs more poorly than the rules learned by LAB, we can see a clear advantage, in this case, in learning rules as opposed to using expert advice. In addition, the resulting rule base, C, is much easier to comprehend than either the decision tree learned by ID3 or the disjuncts returned by PFOIL. An example of the concept `right internal capsule` learned by each system is shown in Figure 4.6. In this figure, the format of the rules learned by LAB is in the form `(← m d)`, instead of `d` $\rightarrow$ `m` as we have been using. Also, in the ID3 tree, a feature value of 0

LAB concept:

```
(<-(Poorokn-Direction Lhoriz)(Right-Internal-Capsule Present))
(<-(Facial-Side-Type Left-Central)
       (Right-Internal-Capsule Present))
(<-(Weakness-Type Hemiparesis-Left)
       (Right-Internal-Capsule Present))
(<-(Decram-Side Left-Severe)(Right-Internal-Capsule Present))
(<-(Disoriented-Degree Mild)(Right-Internal-Capsule Present))
(<-(Facenumb-Side Left)(Right-Internal-Capsule Present))
```

ID3 Concept:

```
((RIGHT-INTERNAL-CAPSULE PRESENT)
 Feature: DECRAM-SIDE
  0 (0.650)
    Feature: TWOPOINT-SIDE
     0 (0.885)
       Class is: -
     LEFT (0.115)
       Feature: DENIAL
        0 (0.333)
          Class is: +
        PRESENT (0.667)
          Class is: -
     RIGHT (0.000)
       Class is: -
  RIGHT-MILD (0.175)
    Class is: -
  RIGHT-SEVERE (0.000)
    Class is: -
  LEFT-MILD (0.150)
    Class is: -
  LEFT-SEVERE (0.025)
    Class is: +)
```

PFOIL Concept:

```
(Right-Internal-Capsule Present)
Positive:
Decram-Side=Left-Severe ∨ (Vibloss-Side=Left ∧ Decloc-Degree=0)
```

Figure 4.6: Example Concepts Learned

indicates that the feature is not present. This tree also illustrates an instance of using information about the absence of a manifestation to make decisions. An example of an entire rule base learned by LAB is given in the appendix.

Finally, we turn to the number of diagnoses returned by the rule bases learned by LAB during training and testing. Ideally, we would like a single, correct diagnosis returned by our abductive mechanism for each example. We have already seen that the goal of getting a correct diagnosis was not met. However, as we have said, the number of diagnosis returned per example is reasonably close to one during training, and closer to 1.5 during testing. This is far better than the expert KB, which produces an average of 4.5 minimum covers per example. There are two reasons why multiple diagnoses may occur in our result. First, we may not learn any rules for some manifestations, in which case their presence in an example does not help us discriminate between the multiple diagnoses generated by the other manifestations. Second, we sometimes learn too many rules for a manifestation, so that all the manifestations in a case are caused by two or more of the same disorders. In this case, if there were more manifestations in the example, it might help discriminate between these diagnoses.

Chapter 5

Related Work

Since no other system learns rules which are used abductively, there are no systems with which to make a direct comparison. However, there are many systems which can learn to do diagnosis, and many methods for reasoning abductively.

Almost any learning system can learn to do diagnosis, some more successfully than others. We have already mentioned the methods of learning rules for deduction, both in the introduction and in our comparisons with ID3 and PFoil. An alternative is Protos (Porter et al., 1990), which uses an exemplar-based approach to learning. Protos has been tested, with favorable results, in the domain of clinical audiology. Two other methods that seem particularly well-suited to do diagnosis are neural networks and Baysian Networks. Neural networks use nodes and links with weights to represent knowledge. Backpropagation, as we have discussed, is the usual technique used to modify the weights during training. While capable of learning to make diagnoses which may include multiple disorders, neural networks have several disadvantages. First, as we saw, they take a long time to train. Second, the learned network is difficult for a domain expert to interpret. This would make it impractical to use in a real world situation, where the diagnostician would typically like to see explanations for the diagnosis output. Our causal rules could handle this much more easily.

A second method is Baysian Networks (Pearl, 1988), which are also networks which represent causality in a graph, but with conditional probabilities and some prior probabilities attached to the nodes. The nodes represent states of affairs (our disorders and manifestations), and the arcs are the causal connections. Some attempts have been made to learn Baysian Networks (Cooper and Herskovits, 1992; Geiger et al., 1990), but they have not been used in a diagnostic domain. Although their probabilistic nature might help these systems perform well in diagnosis, they have trouble, used on their own, in coping with the presence of multiple disorders. The reason for this is that the information they provide is the probability that each disorder is present. A question that arises is how high the probability should be to decide whether a disorder is actually present. Because of this, Baysian Networks may have to be integrated with another system to help discriminate between diagnoses. Also, actually performing diagnosis once the network is built is in general NP-hard if loops exist in the network.

Many alternative formalisms exist for abduction. (Allemang et al., 1987) use an abductive model similar to that of Peng & Reggia, but with an approximate algorithm. The ATMS (Assumption-based Truth Maintenance System) (de Kleer, 1986) reasons using propositional Horn-clause axioms. The ATMS and similar systems have been used in domains other than diagnosis. Recently, Levesque (Levesque, 1989) has shown that with a slightly different formalism of abduction than used here, the ATMS performs abduction precisely as he defines it. The ACCEL system by Ng (Ng, 1992) extends the ATMS to first-order Horn-clause axioms with variables, and also works in domains other than diagnosis. It also uses several efficiency measures while performing the abduction. Because of these systems' utility in multiple domains, such as

natural language understanding, we foresee that our approach could eventually be extended to learning in other domains as well.

Finally, there are alternatives to using abduction for diagnosis. First, model-based diagnosis (de Kleer and Williams, 1987; Reiter, 1987) uses a model to predict the behavior of a system, and to make a diagnosis based on the discrepancy, if any, between the predictions of the model and the actual behavior of the system. One weakness in the method is that the model can be wrong or inexact. A second alternative is Reiter's (Reiter, 1987) diagnosis from first principles, which reasons from the system description and observations of the systems behavior.

Chapter 6

Future Work

There are many opportunities for future work in this area. We can see some of these in the results obtained with LAB.

First, we believe that our training accuracy is not at the maximum level possible, even with the presence of inconsistent examples. There are several modifications we could make to improve this. First, we could use different or additional heuristics during our hill-climbing search. One possibility is to use better measures of the utility of adding a rule throughout the search. Second, we could perform search with backtracking to have a better chance of finding the most accurate rule base possible. Our original attempts at an algorithm to achieve our goal were along these lines, but were met with limited success because we did not have the focussed goal of finding the maximum possible accuracy. Third, we could use a bias other than the minimum cardinality when choosing an answer from BIPARTITE. As pointed out in (Tuhrim et al., 1991), minimum cardinality is not always the most successful criterion for choosing an explanation. For example, sometimes choosing a non-minimum cover leads to a more likely diagnosis than a minimum cover.

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 using a probabilistic model for our abduction. One such model is outlined in (Peng and Reggia, 1990), and ranks explanations based on

information about the probability of a disorder occurring and the probability that a disorder *causes* a manifestation. These are assumed to be available, as is the causation relation for BIPARTITE. This is a more realistic assumption than traditional probabilistic approaches, which require information about the frequency with which disorders and manifestations occur together. This is much more difficult to obtain, and causes computations to become intractable. In order to use these probabilities in our approach, investigation is needed as to how they might be learned.

Another possible solution to the multiple diagnosis problem is to assure that at least one rule is associated with each manifestation occurring in the training examples. This does not always happen now because of hill-climbing. This way, each manifestation will be guaranteed to help discriminate between competing diagnoses during testing. Another possibility is to learn more general rules for each disorder. For example, if damage in the `right internal capsule` may cause a disorientation of degree mild, it is likely to cause a disorientation of moderate, or even severe degree. This could be useful for helping LAB deal with manifestations unseen during training.

Third, the answers returned during testing are overly general, as can be seen in the results on specificity. Again, probabilistic measures might help, by ruling out unlikely disorders. Also, we could compare and combine the answers given by BIPARTITE with those of another system, perhaps to narrow the number of disorders hypothesized. However, as we have noted, we would rather have an overly general diagnosis than one which was missing disorders.

Fourth, our training time is too slow. We could improve this by using methods other than always running BIPARTITE to choose which rules to add to C. Again, heuristics and a more intelligent, faster search could improve

the training time. Also, we could use domain specific knowledge to help focus the search, reducing search time as well as potentially improving the accuracy. For example, in our brain damage domain, we could use the knowledge that damaged areas are typically located near each other to rule out implausible diagnoses.

Additional experiments are required to more clearly determine the advantages and disadvantages of learning for abduction. Experiments in other domains are desirable; however we know of no other existing data sets for multiple-disorder diagnosis. Experiments on some of the standard single-disorder data sets should be performed; however, abduction's real advantage is on multiple-disorder diagnosis. Also, LAB should be compared to additional learning methods such as PROTOS, instance-based methods (Aha et al., 1991), etc.

Finally, 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 first-order predicates (Ng, 1992).

Chapter 7

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, as well as one that is more sensitive and comprehensible than concepts learned by ID3.

This thesis has made several important contributions:

1. We have demonstrated a method for learning rules which will be used abductively by building a system, LAB.

2. We have shown that learning for abduction is more successful at predicting which disorders are present than two other systems and a knowledge base built by domain experts. This success was shown for a real world data set.

3. We have introduced a new paradigm of learning. Learning for abduction is a previously untried method with much promise.

In summary, this thesis has demonstrated via an implemented system that learning for abduction is a possible and successful paradigm of learning. This method has much potential for exploration and experimentation, and the future holds much promise for this approach to the learning task.

Appendix A

Abbreviations

Abbreviations used in Section 4.3:

Disorders:

`lfl`: left frontal lobe,

`lic`: left internal capsule,

Manifestations:

`at(hr)`: abneom type = hgaze right,

`fst(rc)`: facial side type = right central,

`ds(mod)`: dysarthria severity = moderate,

`wt(hr)`: weakness type = hemiparesis left,

`ts(r)`: tongweak side = right,

`ded(d)`: decloc-degree = drowsy,

`des(rmi)`: decram side = right mild,

`as(ri)`: abndtrs side = right incdtr,

`bs(r)`: babs side = right.

Appendix B

Sample Rules

Sample rule base learned by LAB:

((←(poorokn-direction lhoriz) (right-frontal-lobe present))

(←(facial-side-type left-central) (right-frontal-lobe present))

(←(weakness-type hemiparesis-left) (right-frontal-lobe present))

(←(babs-side left) (right-frontal-lobe present))

(←(dss-side left) (right-frontal-lobe present))

(←(dysarthria-severity mild) (right-frontal-lobe present))

(←(decram-side left-mild) (right-frontal-lobe present))

(←(abndtrs-side left-incdtr) (right-frontal-lobe present))

(←(facenumb-side right) (left-internal-capsule present))

(←(weakness-type hemiparesis-right) (left-internal-capsule present))

(←(decram-side right-mild) (left-internal-capsule present))

(←(abndtrs-side right-incdtr) (left-internal-capsule present))

(←(babs-side right) (left-internal-capsule present))

(←(gait-type other) (left-internal-capsule present))

(←(pp-side right-mild) (left-internal-capsule present))

(←(vibloss-side right) (left-internal-capsule present))

(←(facenumb-side right) (left-pons present))

(←(weakness-type hemiparesis-right) (left-pons present))

(←(ataxia-type limb-right-mild) (left-pons present))

(←(decram-side right-mild) (left-pons present))

(←(babs-side right) (left-pons present))

(←(gait-type unsteady) (left-pons present))

(←(facial-side-type right-central) (left-pons present))

(←(dysarthria-severity mild) (left-pons present))

(←(ataxia-type limb-left-mild) (left-pons present))

(←(decram-side left-mild) (left-pons present))

(←(pp-side right-mild) (left-pons present))

(←(posloss-side right) (left-pons present))

(←(disoriented-degree mild) (left-frontal-lobe present))

(←(poorokn-direction rhoriz) (left-frontal-lobe present))

(←(abneom-type hgaze-right) (left-frontal-lobe present))

(←(facial-side-type right-central) (left-frontal-lobe present))

(←(tongweak-side right) (left-frontal-lobe present))

(←(weakness-type hemiparesis-right) (left-frontal-lobe present))

(←(babs-side right) (left-frontal-lobe present))

(←(dysarthria-severity moderate) (left-frontal-lobe present))

(←(abndtrs-side right-incdtr) (left-frontal-lobe present))

(←(decloc-degree drowsy) (left-frontal-lobe present))

(←(decram-side right-mild) (left-frontal-lobe present))

(←(poorokn-direction lhoriz) (right-internal-capsule present))

(←(facial-side-type left-central) (right-internal-capsule present))

(←(weakness-type hemiparesis-left) (right-internal-capsule present))

(←(decram-side left-severe) (right-internal-capsule present))

(←(ataxia-type limb-right-mild) (right-cerebellar-hemisphere present))

(←(decram-side right-mild) (right-cerebellar-hemisphere present))

(←(gait-type unsteady) (right-cerebellar-hemisphere present))

(←(abneom-type fourn-right) (right-midbrain present))

(←(ataxia-type limb-left-mild) (right-midbrain present))

(←(decram-side left-mild) (right-midbrain present))

(←(babs-side left) (right-midbrain present))

(←(facial-side-type right-central) (left-internal-capsule present))

(←(dysarthria-severity moderate) (left-internal-capsule present))

(←(nonfluency-severity mild) (left-frontal-lobe present))

(←(repetition-severity mild) (left-frontal-lobe present))

(←(dysarthria-severity mild) (left-internal-capsule present))

(←(vf-deficit-side-type left-hemianopsia) (right-parietal-lobe present))

(←(vf-deficit-side-type right-hemianopsia) (left-parietal-lobe present))

(←(nonfluency-severity severe) (left-temporal-lobe present))

(←(abneom-type hgaze-left) (right-temporal-lobe present))

(←(compdef-severity severe) (left-frontal-lobe present))

(←(hemineglect-side right) (left-temporal-lobe present))

(←(facenumb-side right) (left-frontal-lobe present))

(←(decloc-degree drowsy) (right-temporal-lobe present))

(←(disoriented-degree mild) (right-frontal-lobe present))

(←(pp-side left-mild) (right-parietal-lobe present))

(←(pp-side right-mild) (left-frontal-lobe present))

(←(anomia-severity mild) (left-thalamus present))

(←(dss-side left) (right-occipital-lobe present))

(←(ataxia-type limb-right-mild) (left-frontal-lobe present))

(←(repetition-severity severe) (left-parietal-lobe present))

(←(pp-side right-moderate) (left-temporal-lobe present))

(←(dysarthria-severity mild) (left-parietal-lobe present))

(←(dysarthria-severity moderate) (right-frontal-lobe present))

(←(decloc-degree drowsy) (right-frontal-lobe present))

(←(decloc-degree stupor) (right-parietal-lobe present))

(←(disoriented-degree mild) (right-internal-capsule present))

(←(disoriented-degree mild) (right-temporal-lobe present))

(←(disoriented-degree mild) (left-internal-capsule present))

(←(facenumb-side left) (right-frontal-lobe present))

(←(facenumb-side left) (right-internal-capsule present))

(←(facenumb-side left) (right-temporal-lobe present)))

Bibliography

Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1):37–66.

Allemang, D., Tanner, M. C., Bylander, T., and Josephson, J. R. (1987). Computational complexity of hypothesis assembly. In *Proceedings of the Tenth International Joint conference on Artificial intelligence*, pages 1112–1117. Milan, Italy.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks.

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.

de Kleer, J. (1986). An assumption-based TMS. *Artificial Intelligence*, 28:127–162.

de Kleer, J. and Williams, B. C. (1987). Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130.

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.

Kassirer, J. (1978). Clinical problem solving: A behavioral analysis. *Annals of Internal Medicine*, 89:245–255.

Konolige, K. (1992). Abduction versus closure in causal theories. *Artificial Intelligence*, 53:255–272.

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.

Michalksi, R., Mozetic, I., Hong, J., and Lavrac, N. (1986). The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 1041–1045. Philadelphia, PA.

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.

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

Peirce, C. S. (1958). *Collected Papers of Charles Sanders Peirce*. Cambridge, Mass.: MIT Press.

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

Pople, Jr., H. E. (1973). On the mechanization of abductive logic. In *Proceedings of the Third International Joint Conference on Artificial Intelligence*, pages 147–152.

Porter, B., Bareiss, R., and Holte, R. (1990). Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence*, 45:229–263.

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.

Quinlan, J. R. (1987). Generating production rules from decision trees. In *Proceedings of the Tenth International Joint conference on Artificial intelligence*, pages 304–307. Milan, Italy.

Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95.

Rubin, A. (1975). The role of hypothesis in medical diagnosis. In *Proceedings of the Fourth International Joint conference on Artificial intelligence*, pages 856–862.

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.

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.

VITA

Cynthia Ann Thompson was born in Kingston, New York, on June 10, 1966, the daughter of Nannette Ray and Ralph Edward Thompson. After completing her work at Independence High School in Charlotte, North Carolina, she entered North Carolina State University in Raleigh, North Carolina. During four separate semesters in college, she worked at NASA/GSFC in Greenbelt, Maryland, as a participant in the cooperative education program. She received the Bachelor of Science in Computer Science from North Carolina State University in May, 1989. During the next two years, she was employed as a consultant with the Information Consulting Group, which became a part of McKinsey and Company during her employment. In August, 1991, she entered the Masters program of the Department of Computer Sciences at the University of Texas at Austin. In the Spring of 1993, she was inducted into The Honor Society of Phi Kappa Phi. Starting in the fall of 1993, she will be enrolled in the PhD program of the Department of Computer Sciences at the University of Texas.

Permanent address: 7909 Meadowdale Lane
Charlotte, North Carolina
28212

This thesis was typeset[1] with LaTeX by the author.

[1] LaTeX document preparation system was developed by Leslie Lamport as a special version of Donald Knuth's TeX program for computer typesetting. TeX is a trademark of the American Mathematical Society. The LaTeX macro package for The University of Texas at Austin thesis format was written by Khe-Sing The.