Theory Revision via Prior Operationalization

Allen Ginsberg
Knowledge Systems Research Department
AT&T Bell Laboratories
Holmdel, NJ 07733

Abstract

Research in machine learning often focuses either on *inductive* learning - learning from experience with minimal reliance on prior theory - or, more recently, on *explanation-based* learning - deducing general descriptions from theories with minimal reliance on experience. *Theory revision* unites these two concerns: one must revise one's theory in the light of experience, but one must simultaneously *use* information implicit in the theory in order to guide the revision process. This paper focuses on the second step of a unified three-step method for solving theory revision problems for certain classes of empirical theories. Test results for the first two phases of this approach are reported.

1 Introduction and Overview

A theory revision problem exists for a theory $\mathcal{T}$ when $\mathcal{T}$ is known to yield incorrect results for given cases in its intended domain of application. The goal of theory revision is to find a revision $\mathcal{T}'$ of $\mathcal{T}$ which handles all known cases correctly, makes use of the theoretical terms used in $\mathcal{T}$, and may, with a reasonable degree of confidence, be expected to handle future cases correctly. In contrast to pure inductive learning from experience, theory revision is not only guided by the information implicit in $\mathcal{T}$, but also attempts to *preserve* the language and, as much as possible, the structure of $\mathcal{T}$.

This paper focuses on the second step of a unified three-step method - integrating aspects of explanation-based learning [Mitchell, Keller, and, Kedar-Cabelli, 1986], inductive learning [Michalski 1983], and heuristic approaches to knowledge base refinement [Ginsberg, Weiss, and Politakis, 1988] - for solving theory revision problems for certain classes of empirical theories. In the first step the theory is "translated" into a form that is more amenable to inductive learning techniques. As we shall see (section 2), this step may be viewed as a *complete prior* "operationalization" of the theory, in the sense of the term employed in explanation-based learning [Mitchell, Keller, and, Kedar-Cabelli, 1986]. This process is called *theory reduction*, and the resulting translation is called the *reduced theory*. The notion of theory reduction is discussed in [Ginsberg, 1988a], and detailed accounts of the application of this idea to "expert system theories" are given in [Ginsberg, 1988b].

The second step involves modifying the reduced theory in order to improve its empirical adequacy. At a high level, the methods described here differ from those described, for example, in [Michalski 1983], in that the basic task is to "tailor" the reduced theory so it "fits the facts," rather than build or rebuild a general description of the facts from the bottom up. This perspective manifests itself in such items as the calculation and use of *theoretical expectations* of correlations between observables and theoretical terms (see section 3.4), as well as the use of certain "conservative" strategies, e.g., attempting to generalize expressions that are already "closest to being satisfied" in a given case (see section 3.1). In contrast to the heuristic approach to knowledge base refinement advocated in [Ginsberg, Weiss, and Politakis, 1988], the *Reduced Theory Learning System* (RTLS) described here does *not* employ a cyclic generate-test-select hill-climbing strategy for discovering efficacious refinements.

Once the reduced theory has been modified, the final step involves a "retranslation" of the modified reduced version back into the entire language of the original theory. This step is necessary because the reduced theory only makes use of a subset of the vocabulary of the original theory. While the parsimony of the reduced theory is desirable in the learning step, it is undesirable as a final goal, since a reduced theory will generally be a less compact and efficient representation for actual use than a theory that uses a richer language and structure. The retranslation step of the method is not discussed here; however, an algorithm for automatic retranslation of expert system theories is known.

2 Theory Reduction & EBL

For a theory $\mathcal{T}$ to have any utility for a system it must be possible for the system to *apply* $\mathcal{T}$ - in essentially mechanical fashion - to problem cases that arise in the domain in question. We may therefore view a useful empirical theory as defining an "inference mechanism" that relates "observable" features of problem cases to "theoretical" entities or processes for which the theory posits certain law-governed behavior.

In view of this, let $\mathcal{T}$ be a theory and let the vocabulary (predicate symbols, propositional constants, etc.) of $\mathcal{T}$ be divided into two disjoint subsets $\mathcal{T}_o$ and $\mathcal{T}_t$. We refer to these as the *observational* (operational) and *theoretical* (non-operational) vocabulary of $\mathcal{T}$, respectively [Nagel, 1961; Keller, 1987]; for the sake of brevity, we will henceforth refer to theoretical terms as *hypotheses*. We may view $\mathcal{T}$ as implicitly specifying a (partial) function, having all possible combinations of items in $\mathcal{T}_o$ as domain and all possible combinations of items in $\mathcal{T}_t$ as range: given some combination of observables as "input," $\mathcal{T}$ will yield some combination of hypotheses as "output," i.e., this is

$\mathcal{T}$'s answer for this case. Complete theory reduction may be seen as a transformation of this implicit function to a set of functions $\mathcal{L}$ more amenable to analysis and revision: for each hypothesis $\tau \in \mathcal{T}_t$, $l(\tau) \in \mathcal{L}$ is a *minimal sum-of-products* (minimal disjunctive normal form) expression in which each product term (disjunct) consists solely of observables. Following the terminology of de Kleer [1986], we say that $l(\tau)$ is the *label* for τ, and each product term in $l(\tau)$ is said to be a (minimal) *environment* for τ. Intuitively, $l(\tau)$ represents all the possible minimal sets of observables that would cause the theory $\mathcal{T}$ to assert τ.

EBL systems may be viewed as involving dynamic partial theory reduction: an "output" of a typical learning episode in EBL systems is generally some product term in $l(\tau)$, for some τ. As each new instance is explained (and generalized) a particular item in the theoretical vocabulary is being "partially reduced" to a set of predicates that meets an *operationality criterion*. For Mitchell *et al.* [1986] the operationality criterion says that the generated generalization must be "expressed in terms of predicates used to describe examples... or other selected, easily evaluated, predicates from the domain theory." This is basically the same as saying that the generated generalization must be entirely in the observational vocabulary.[1]

As new instances are presented to an EBL system, new explanations (inherent in the theory) will be generated and generalized. For finite propositional theories there can be only a finite number of instances (perhaps very large) that yield new explanations and generalizations. [2] In the limit, when all the instances are seen, the domain theory will be completely operationalized. Complete theory reduction yields the same results as exhaustive operationalization, but does not require the actual presentation of cases to achieve them: in this sense, the theory is exhaustively operationalized *prior* to case presentation.

2.1 Reduction of Expert System Theories

We consider an *expert system theory* $\mathcal{E}$ to be a *restricted* propositional logic theory. That is, $\mathcal{E}$ consists of a set of conditionals in propositional logic, i.e., the rules or knowledge base. A sentence $\alpha \rightarrow \beta$ is considered to follow from $\mathcal{E}$ iff, to put it loosely, β can be derived from α and $\mathcal{E}$ via a sequence of applications of a generalized version of modus ponens. $\mathcal{E}$ is said to be *acyclic* if, roughly speaking, a sentence of the form $\alpha \rightarrow \alpha$ does not follow from $\mathcal{E}$.

A two-step algorithm for the complete prior reduction of acyclic expert system theories, and a system, *KB-Reducer*, that implements the algorithm are discussed in [Ginsberg, 1988b]. In the first step the rules in $\mathcal{E}$ are partitioned into disjoints sets called *rule levels*. A rule r is in level 0 iff the truth-value of the left-hand side of r is a function of the truth-values of observables only. A rule r is in level n, iff the truth-value of the left-hand side of r is a function of the truth-values of observables and hypotheses that are concluded only by rules at levels $0, \ldots, n-1$. This partition defines a partial-ordering for computing the reduction of all hypotheses: each rule in level 0 is processed (exactly once), then each rule in level 1, etc.

KB-reducer has been used to reduce several knowledge bases. Knowledge bases of approximately 50 and 150 rules in size were reduced in 40 cpu seconds and 5 cpu minutes respectively. The early rheumatology knowledge base [Lindberg *et al.*, 1980] (call it "Rheum" for short) which was used to conduct the experiments reported below (section 4) has 84 observables, 73 hypotheses, and 367 rules when translated into the rule language used by KB-Reducer. There are 4 rules levels. The total cpu time to compute the reduction was approximately 10 hours on a TI Explorer[3] II. The total number of environments in the labels is 34,522.

3 Refinement of the Reduced Theory

This section describes the methods employed by an inductive learning program, RTLS, that takes as its input a set of labels $\mathcal{L}$ and a set of cases $\mathcal{C}$ such that for each $c \in \mathcal{C}$, Answer(c), the "correct answer" for case c, is known. For the sake of brevity, we use the following notation and terminology. (Table 1 below summarizes most of the notation used in this section.) Let $l(\tau) \in \mathcal{L}$ represent the label (at some specified point in the training process) for hypothesis τ. The (current version of the) theory would assert τ in case c iff $l(\tau)$ is satisfied by c, i.e., there is some environment in $l(\tau)$ that is satisfied by (contained in) c. For a given c, the set of hypotheses τ such that $l(\tau)$ is satisfied by c will be referred to as the "outcome vector" for c. The phrases "τ-case" and "non-τ-case" refer, respectively, to any member of $\mathcal{C}$ which is such that Answer(c) includes, or does not include τ. Suppose that $l(\tau)$ fails to have any of its environments satisfied in a τ-case c. In this event we say that τ and $l(\tau)$ "require generalization" and that c "poses a generalization problem" for $l(\tau)$ and τ. Suppose that $l(\tau)$ has one or more of its environments satisfied in some non-τ-case c. In this case we say that τ and $l(\tau)$ "require specialization" and that c "poses a specialization problem" for $l(\tau)$ and τ.

RTLS currently uses a five-phase procedure in refining reduced theories. The first two phases involve *massive* label generalization and specialization; the third and fourth phases involve *focused* label generalization and specialization; the fifth phase corrects any problems that are not corrected in the first four phases. In massive label refinement one attempts to counteract "systematic" errors in a theory by deleting or adding observables to relatively large numbers of environments in an effort to match observed correlations, without trying to correct any specific problem. As we shall see, this is a useful tactic when revising theories whose reductions are much larger than the number of cases in $\mathcal{C}$. In focused label refinement specific

[1] DeJong and Mooney [1986] point out problems with the notion of a well-defined, fixed observation language and Keller [1987] argues for a notion of operationality that is more closely tied to the "objectives" of the performance system. For the *sorts of theories considered in this paper*, however, it is reasonable, both from a formal point of view, and from the point of view of the typical intended application domains, e.g., medical diagnosis, to view theories as having a fixed observational vocabulary over substantial periods of time.

[2] This will also be true for predicate logic theories with respect to their finite models [Ginsberg, 1988a].

[3] Trademark of Texas Instruments Inc.

$\mathcal{T}_o$, o	the observational terms (observables) of $\mathcal{T}$ (the given theory); a variable over these
$\mathcal{T}_t$, τ	the theoretical terms (hypotheses) of $\mathcal{T}$; a variable over these
$\mathcal{C}$, c	the given set of cases; a variable over individual cases
Answer(c)	the given (correct) theoretical description for c (may contain several hypotheses)
$l(\tau)$	the label for τ (initially determined by reducing $\mathcal{T}$)
$\mathcal{L}$	the set of $l(\tau)$ for $\tau \in \mathcal{T}_t$
outcome vector	for a given c, this is the set of τ whose $l(\tau)$ are satisfied by c
τ-case	a c whose Answer(c) includes τ
non-τ-case	a c whose Answer(c) does not include τ
generalization problem	for a τ, is a τ-case c that does not satisfy $l(\tau)$
Gen($\mathcal{C}, \tau$); g	the set of τ-cases $\subset \mathcal{C}$ posing generalization problems for τ; a variable over these
e	a variable over environments
$e - n$	the result of removing n specified observables from e, so that $e - n$ is satisfied by given g
G_n	the set of $g \in$ Gen($\mathcal{C}, \tau$), for which a (non-empty) $e - n$ exists
E_n	for a $g \in G_n$, the set of all $e - n$ for g
S	for a $e - n \in E_n$, the set of non-τ-cases satisfying $e - n$
S-patches	for some $e - n$, the set of all o such that o is true in g, but not in any S
$e - n + o$	the result of adding some $o \in S$-patches to $e - n$
specialization problem	for a τ, is a non-τ-case c that does satisfy $l(\tau)$
Spec($\mathcal{C}, \tau$); s	the set of non-τ-cases $\subset \mathcal{C}$ posing specialization problems for τ; a variable over these
Spec-envs(s, τ)	the set of $e \in l(\tau)$ satisfied by (non-τ-case) s
Left-Spec-envs(τ)	the union over $s \in$ Spec($\mathcal{C}, \tau$) of all the environments that remain in Spec-envs(s, τ) after certain specialization procedures have been applied
New-Gen(τ)	cases that would become generalization problems for τ if all environments in Left-Spec-envs(τ) were deleted from $l(\tau)$

Table 1: Important Symbols and Terminology Defined

additions, deletions, and modifications are made to targeted environments in labels in ways that are guaranteed to correct specific problem cases. A key pair of principles employed throughout these procedure is: whenever solving a generalization {specialization} problem be sure not to create new specialization {generalization} problems. A simple schematic example of the methods discussed here is given in figure 1 below.

3.1 Focused Label Generalization

Let Gen($\mathcal{C}, \tau$) be the set of τ-cases in $\mathcal{C}$ which pose generalizations problems for τ; and let g be a variable over individual cases in this set. Let G_1 be that subset of Gen($\mathcal{C}, \tau$) consisting of every g for which there exists at least one environment $e \in l(\tau)$ that would be satisfied in g if exactly *one* observable were to be deleted from e. That is, removal of the observable in question from e would yield an environment for τ that is satisfied in g. RTLS will initially try to correct the generalization problems posed by G_1. Any $g \in G_1$ whose generalization problem is solved in this phase is removed from Gen($\mathcal{C}, \tau$). RTLS will then move on to consider $G_2 \subset$ Gen($\mathcal{C}, \tau$) which is defined in similar fashion to G_1. That is, for any g in G_2 there exists at least one environment $e \in l(\tau)$ that would be satisfied in g if exactly *two* observables were to be deleted from e. This process continues until all the generalization problems posed by Gen($\mathcal{C}, \tau$) are solved or the number of observables that would have to be deleted equals the length of the largest environment in $l(\tau)$.[4]

Let us suppose that for some n, RTLS is currently concerned with the cases in G_n; let g be a member of G_n. Let $E_n \subset l(\tau)$ be the set of $e \in l(\tau)$ that would be satisfied in g if n of e's observables were removed. For each $e \in E_n$, RTLS forms $e - n$, i.e., e with the n observables which make it unsatisfied in g, removed. For each such $e - n$ RTLS then determines whether it is satisfied in any *non-τ*-case. If some of the $e - n$'s are *not* satisfied in any non-τ-case, then these environments are added to the label for $l(\tau)$; the generalization problem for g is solved.

Suppose, on the other hand, that each of the $e - n$'s is satisfied in at least one *non - τ* -case. Let S be the set of non-τ-cases satisfied by one of these environments. For each $e - n$ and its associated S, RTLS will try to find *all* the observables o which are such that o is true in g but not true in any of the cases in S; let us call the set of such o's, the *S-patches* for $e - n$. Adding any $o \in S$-patches to $e - n$ produces an environment - let us designate it by the notation $e - n + o$ - that will be satisfied in g but not satisfied in any non-τ-case. If S-patches exists for *any* $e - n$, then for *every* $e - n$ that has S-patches, RTLS will add $e - n + o$ to $l(\tau)$ for *every* o in the S-patches for $e - n$.

If, on the other hand, S-patches does not exist for any $e - n$, RTLS will move on to try something called *theory-driven label generalization*. For each $e - n$ a sorted list of *candidates* is formed, as follows. We consider every o true in g that is not in $e - n$. The list of such o's is sorted in decreasing order by the *theoretical expectation* of their "correlation strength" with respect to τ - a number given

[4] The idea of first trying to generalize environments that are "closest to being satisfied" in a case is analogous to the idea of trying to generalize rules closest to being satisfied used in [Politakis and Weiss, 1984; Ginsberg, Weiss, and Politakis, 1988].

Theory	Labels of Reduced Theory
$ab \vee ac \vee bc \rightarrow \tau_1$	$\tau_1 = ab \vee ac \vee bc$
$e \vee f \vee g \rightarrow \tau_2$	$\tau_2 = e \vee f \vee g$
$d\tau_1\tau_2 \rightarrow \tau_3$	$\tau_3 = abde \vee acde \vee bcde \vee abdf \vee acdf \vee bcdf \vee abdg \vee acdg \vee bcdg$

After Massive Refinement	After Focused Refinement
$\tau_1 = a \vee b$	$\tau_1 = a \vee b \vee c$
$\tau_2 = e \vee f \vee g$	$\tau_2 = ef \vee eg \vee fg$
$\tau_3 = ae \vee be \vee af \vee bf \vee adg \vee bdg$	$\tau_3 =$ *unchanged*

Cases

Case:	a	b	c	e	f	g	fg	ef	eg
Answer:	τ_1	τ_1	τ_1	nil	nil	nil	τ_2	τ_2	τ_2

Case:	abg	bfg	aef	$abef$
Answer:	τ_1	$\tau_1\tau_2\tau_3$	$\tau_1\tau_2\tau_3$	$\tau_1\tau_2\tau_3$

Figure 1: A Simple Example

by the percentage of environments in $l(\tau)$ containing o (see section 3.4 below).[5] Examining the candidate list in sorted order, RTLS tries to find a subset $O = \{o_1, ..., o_k\}$ of the candidates which is such that each $o_i \in O$ has positive correlation strength with τ, and such that the conjunction of observables of O is false in every non-τ-case. If such an O is found, it is added to $e - n$, and this new environment is added to $l(\tau)$; the generalization problem for g is solved. If such an O is not found, then the problem posed by g will be reconsidered by RTLS when it looks at environments in $l(\tau)$ with exactly $n+1$ observables removed, i.e., at the G_{n+1} phase.

3.2 Focused Label Specialization

Let Spec($\mathcal{C}, h$) be the set of *non-τ-cases* in $\mathcal{C}$ which pose specialization problems for τ; and let s be a variable over individual cases in this set. For each $s \in$ Spec($\mathcal{C}, h$), let Spec-Envs(s, h) be the set of environments in $l(\tau)$ satisfied in case s. In order to solve the specialization problem posed by s, *every* environment e in Spec-Envs(s, h) must be modified - or, if necessary, deleted from $l(\tau)$ - in such a way that e no longer is satisfied in s. Once again, however, RTLS will make a modification to solve a specialization problem, only if doing so does not result in a new generalization problem coming into existence.

Let e be an environment in Spec-Envs(s, h). RTLS will first attempt to *add* observables to e to prevent it from being satisfied in s, without causing it to become unsatisfied in any τ-case it which it may currently be satisfied. It may be possible to do this in one of two ways. First RTLS tries to find an observable o satisfied in *every* τ-case but not satisfied in *any* non-τ-case. If such an o exists replacing e in $l(\tau)$ with the environment $e + o$ will contribute to solving the specialization problem - it will solve it only if e is the only member of Spec-Envs(s, h) - without generating new generalization problems. In fact, if there are several such observables $o_1, \ldots, o_n$, RTLS will add each $e + o_i$ to $l(\tau)$ for $1 \leq i \leq n$. If such an o does not exist, RTLS will try the tactic in reverse: find an o true in *every* non-τ-case that is *not* true in any τ-case. Again, for every such o, the environment $e + \bar{o}$, i.e., e with the *negation* of o added to it - is added to $l(\tau)$ (and e, of course, is removed). If either of these tactics is successful, RTLS will remove e from Spec-Envs(s, h); if Spec-Envs(s, h) is now empty the specialization problem for s is solved.

Suppose that RTLS has tried these procedures but Spec-Envs(s, h) is still not empty. Let Left-Spec-Envs(h) be the *union* of Spec-Envs(s, h) for every s after the preceding tactics have been attempted. Intuitively, if all the environments in Left-Spec-Envs(h) were simply removed from $l(\tau)$ all remaining specialization problems for τ would be solved; the problem, of course, is that new generalization problems might thereby be created. Let New-Gen(h) be the (possibly empty) set of cases that would pose new generalization problems for τ if all the environments in Left-Spec-Envs(h) were to be deleted from $l(\tau)$. For each case $g \in$ New-Gen(h), RTLS will attempt to use theory-driven label generalization - see section 3.1 above - to generate a new environment for τ that will be satisfied in g but not generate new specialization problems. If this can be done for every case in New-Gen(h) then deletion of the environments in Left-Spec-Envs(h) from τ - together with addition of the environments generated to handle New-Gen(h) - will solve all specialization problems for τ. However, outright deletion of environments from a label is a tactic that RTLS would prefer to use only as a last resort.

The alternative is to try *theory-driven label specialization* on the environments in Left-Spec-Envs(h). For each $e \in$ Left-Spec-Envs(h) a sorted list of *candidates* is formed, as follows. We consider every o that is not already contained in e. The list of such o's is sorted in decreasing order by the theoretical expectation of their correlation strength with respect to τ (see section 3.4 below). Examining the candidate list in sorted order, RTLS tries to find a subset $O = \{o_1, ..., o_k\}$ of the candidates such that each $o_i \in O$ has positive correlation strength with τ, and such that the conjunction of observables of $e + O$ is false in every non-τ-case. If such an O can be found for every e in Left-Spec-Envs(h), and if replacing e with $e + O$ results in an $l(\tau)$ that does not have new generalization problems, then RTLS will perform these replacements. If this is not the case, and New-Gen(h) is either empty or can be successfully treated by theory-driven label generalization, then Left-Spec-Envs(h) will simply be removed from $l(\tau)$. However, it is possible for theory-driven generalization to fail on New-Gen(h) and for theory-driven specialization to fail on Left-Spec-Envs(h); in that event no action will be taken, and some specialization problem for τ will remain unsolved.

[5] While this idea is similar to Davis's [1979] notion of a *rule model* - which encoded the degree of correlation between the occurrence of items in the antecedents of *rules* with the hypotheses asserted by those rules - it is more general in the sense that a theory can imply a correlation between o and τ, even if they never occur together in any rules.

3.3 Solving All Problems

Suppose that the above procedures have been applied with less than 100% success. Let $\mathcal{L}_g$ be the set of labels for which generalization problems still exist; let $\mathcal{L}_s$ be the set of labels for which specialization problems still exist. The specialization problems are addressed first as follows. Suppose s is a case which poses a specialization problem for label $l(\tau)$: simply delete every environment in $l(\tau)$ that is satisfied in s. We do this for every case that poses a specialization problem for any label. Let $\mathcal{L}'$ be the set of labels *after* all these deletions have been performed. We must now *recalculate* the outcome vector for *every* case c - using $\mathcal{L}'$ - before continuing the training process. This is necessary because the deletions just performed may give rise to new generalization problems which we must address in the next step of the process. Note however, that at this point *all* specialization problems have been solved.

Now we address the remaining generalization problems. Let τ be a hypothesis for which generalization problems exist. Note that an easy way of correcting all these problems for τ - without generating any new problems (assuming that $\mathcal{C}$ is consistent) - is simply to add c to $l(\tau)$ for each case $c \in \text{Gen}(\mathcal{C}, \tau)$, i.e., the set of cases still posing a generalization problem for τ. A better procedure is to *safely* generalize each such c before adding it to $l(\tau)$. This is done by first forming a subvector of c that contains only those observables in c that have positive *theoretical* expected correlation strength with respect to τ. Let c_τ represent this subvector of observables. Now for every *non-τ*-case, c', in which c_τ is satisfied we find an observable $o \in c - c_\tau \& o \notin c'$ such that o has maximum expected *empirical* correlation strength with respect to τ. For each such *non-τ*-case, c', we add the corresponding o to c_τ. The resulting vector of observables is guaranteed to be satisfied in case c but unsatisfied in every *non-τ*-case in $\mathcal{C}$; it now is added to $l(\tau)$. Once this procedure is repeated for every $c \in \text{Gen}(\mathcal{C}, \tau)$, all generalization problems for τ will be solved, and no new problems will be generated.

3.4 Massive Label Refinement

As we have seen, focused label refinement can guarantee that all cases in $\mathcal{C}$ are handled correctly by the refined reduced theory. Recall however, that a good solution to a theory revision problem should yield a rational expectation of general improvement over future cases and not merely over known cases. The greater the expected improvement over the entire domain of cases, the better a solution one has obtained. One can identify situations in which focused label refinement alone will clearly fail to generate the highest expectation of such a general improvement. This will almost certainly be the case when the reduced theory contains a far greater number of environments than the number of cases in $\mathcal{C}$, and $\mathcal{C}$ contains a fairly representative set of cases for the domain.

To appreciate this point, consider the following example. Suppose a label $l(\tau)$ contains 1000 environments for τ, and that *all* of them contain a particular observable o. Suppose that in reality this is a particularly egregious systematic error: o should be in only 100 environments for τ. Suppose further that the training set contains 100 τ-cases, 10 of which contain o. Clearly in this case it is highly unlikely that the focused label refinement process will result in the deletion of o from 900 environments in $l(\tau)$. It is, therefore, highly likely that some τ-cases in the domain but not in $\mathcal{C}$ will (when they become known) pose generalization problems for the new label.

Massive label refinement is an attempt to address the problem of refining theories that have a large reduced form relative to the number of known cases. Massive label refinement involves trying to make the "correlation strength" between observables and hypotheses *implicit* in a theory match the correlation strengths that are *actually observed* in the training cases. (Currently RTLS deals only with first-order correlations, i.e., correlations between a single observable and a single hypothesis.) Thus in the example just given, the fact that *all* the environments for τ contain o raises the *theoretical* expectation that τ and o will *always* occur together. But the 90 τ-cases in which o does not occur, as opposed to the 10 τ-cases in which it does, raises the *empirical* expectation that τ and o occur together 10% of the time. To make the former expectation (quantitatively) match the latter, one should *attempt* to remove o from 90% of the environments in $l(\tau)$; we say "attempt" because removal of o from environment e should only be performed if the resulting environment is not satisfied by any *non-τ* case.

In the case just given one attempts to *decrease* the theoretical expectation for a correlation between an observable and a hypothesis by removing the observable from a certain fraction of the environments in which it occurs in a label. This is *massive label generalization.* If one reverses the example - suppose the theoretical expectation is that o and τ never occur together while the empirical expectation is that they always do - then by a similar argument one is led to the idea of *massive label specialization*: one attempts to *increase* the theoretical expectation for a correlation between an observable and a hypothesis by adding the observable to a certain fraction of the environments in which it does not already occur in a label. (Again, o will not be added to an environment for τ if doing so causes some τ-case to become a new generalization problem).

RTLS attempts massive label refinement prior to focused label refinement. Currently RTLS determines that a label $l(\tau)$ requires *massive label generalization* if the following is true: there are generalization problems for $l(\tau)$, and the percentage of environments in $l(\tau)$ that are *not* satisfied in *any* τ-case is greater than a user specifiable value, currently set at 5%. RTLS determines that $l(\tau)$ requires *massive label specialization* if the following is true: there are specialization problems for $l(\tau)$, and the percentage of environments in $l(\tau)$ that *are* satisfied in *at least one non-τ*-case is greater than a user specifiable value, currently set at 5%.

RTLS will attempt to decrease {increase} theoretical expectations of correlation strengths for any o for which the difference between the implied {observed} correlation and the observed {implied} correlation exceeds a user specifiable value, currently set at 1%. Once massive label adjustment is completed, one must recalculate the outcome vector for every case, and recompute the theoretical expectation for the correlation of every o - τ pair.

The experiments conducted todate indicate that massive label refinement can have the desired impact. In the experiments reported here (section 4) it was observed that performing massive label refinement prior to focused label refinement generally resulted in a 3-5% increase in performance over test cases - cases not included in C - than simply using focused label refinement alone.

4 Empirical Evaluation of RTLS

RTLS has been implemented in common lisp and runs on a Texas Instruments Explorer[6] II. The system has been tested using the aforementioned Rheum knowledge base (section 2.1). A total of 121 cases were available. Initially Rheum misdiagnoses 33 cases: 11 false positives and 22 false negatives. While multiple hypotheses were allowed in Answer(c) for these cases, there is always one of them that is distinguished as the *preferred diagnosis.* As in previous work with Rheum [Politakis and Weiss, 1984; Ginsberg, Weiss, and Politakis, 1988] a case c was judged to be correctly diagnosed by theory $\mathcal{T}$ iff the preferred diagnosis for c had the highest confidence factor of any hypothesis among those reached by $\mathcal{T}$ in case c.

In a typical RTLS-Rheum experiment anywhere from 70 to almost 100 percent of the cases are randomly chosen as training cases and the rest left out for independent testing. Average training time per trial is about 7-10 cpu minutes. The system always trains to 100% correct over the training set. The average performance on the testing set in these trials is nearly always in the 90% to 100% range - which represents improvements ranging from 17 to 27 percent over the initial theory. (It should be noted that in all but a handful of several hundred such experiments, the fifth phase of the procedure, described in section 3.3, did *not* have to be invoked.) Using the more accurate *leave-one-out* method [Lachenbruch, 1967] - which in this case involves running 121 trials, using a single different case as the testing set on each trial, then summing the results - an estimated error rate of 6.7% was obtained. When massive label refinement is *not* used leave-one-out yields an estimated error rate of 11.6%.

5 Conclusion

The results reported here indicate that the basic approach is a feasible and robust solution to the theory revision problem for non-trivial medium size expert system theories. For large scale problems it will undoubtedly be necessary to employ heuristic strategies [Ginsberg, 1986] in order to pinpoint selected portions of the theory for reduction or partial reduction.

6 Acknowledgments

I thank Sholom Weiss and Casimir Kulikowski for access to the rheumatology knowledge base and cases used in these experiments. I thank Keith Williamson for useful discussions related to this topic. I also thank the anonymous reviewers of this paper for useful criticisms.

References

[Davis, 1979] R. Davis. Interactive transfer of expertise: acquistion of new inference rules. *Artificial Intelligence,* 12:121–157, 1979.

[DeJong and Mooney, 1986] G. DeJong and R. Mooney. Explanation-based learning: an alterntive view. *Machine Learning,* 1:145–176, 1986.

[deKleer, 1986] Johan de Kleer. An assumption-based tms. *Artificial Intelligence,* 28:127–162, 1986.

[Ginsberg, 1988a] A. Ginsberg. Theory reduction: operationalization as a prelude to learning. In *Proceedings of AAAI Spring Symposium Series,* Stanford U., 1988.

[Ginsberg, 1988b] A. Ginsberg. Knowledge-base reduction: a new approach to checking knowledge bases for inconsistency and redundancy. In *Proceedings of Seventh National Conference on Artificial Intelligence,* Minneapolis, 1988.

[Ginsberg, 1986] A. Ginsberg. *Refinement of Expert System Knowledge Bases: A Metalinguistic Framework for Heuristic Analysis.* PhD thesis, Department of Computer Science, Rutgers University, 1986.. Forthcoming as *Automatic Refinement of Expert System Knowledge Bases,* Pitman Research Notes in Artificial Intelligence, Pitman Press, London, 1988.

[Ginsberg, Weiss, and Politakis, 1988] A. Ginsberg, S. Weiss, and P. Politakis. Automatic knowledge base refinement for classification systems. *Artificial Intelligence.* To appear in 1988.

[Keller, 1987] R. Keller. Defining operationality for explanation-based learning. In *Proceedings of the Sixth National Conference on Artificial Intelligence,* Seattle, Wa., 1987.

[Lachenbruch, 1967] P. Lachenbruch. An almost unbiased method of obtaining confidence intervals for the probability of misclassification in discriminant analysis. *Biometrics,* 24:639–645, December 1967.

[Lindberg *et al.*, 1980] D. Lindberg, G. Sharp, L. Kingsland, S. Weiss, S. Hayes, H. Ueno, and S. Hazelwood. Computer-based rheumatology consultant. In *Proceedings of the Third World Conference on Medical Informa tics,* pages 1311–1315, North-Holland, 1980.

[Michalski 1983] R. Michalski. A theory and methodology of Inductive learning, *Artificial Intelligence,* 20:111–161, 1983.

[Mitchell, Keller, and, Kedar-Cabelli, 1986] T. Mitchell, R. Keller, and S. Kedar-Cabelli. Explanation-based generalization: a unifying view. *Machine Learning,* 1:47–80, 1986.

[Nagel, 1961] E. Nagel. *The Structure of Science.* Harcourt, Brace, and World, New York, 1961.

[Politakis and Weiss, 1984] P. Politakis and S. Weiss. Using empirical analysis to refine expert system knowledge bases. *Artificial Intelligence,* 22:23–48, 1984.

[6] Trademark of Texas Instruments Inc.