To appear in the
Proceedings of the Fourteenth International Conference on Artificial Intelligence (IJCAI-95),
Montreal, August 1995.

The Complexity of Theory Revision

Russell Greiner*
Siemens Corporate Research   Princeton, NJ 08540-6632
greiner@scr.siemens.com

Abstract

A knowledge-based system uses its database (a.k.a. its "theory") to produce answers to the queries it receives. Unfortunately, these answers may be incorrect if the underlying theory is faulty. Standard "theory revision" systems use a given set of "labeled queries" (each a query paired with its correct answer) to transform the given theory, by adding and/or deleting either rules and/or antecedents, into a related theory that is as accurate as possible. After formally defining the theory revision task and bounding its sample complexity, this paper addresses the task's computational complexity. It first proves that, unless $P = NP$, no polynomial time algorithm can identify the optimal theory, even given the exact distribution of queries, except in the most trivial of situations. It also shows that, except in such trivial situations, no polynomial-time algorithm can produce a theory whose inaccuracy is even close (*i.e.*, within a particular polynomial factor) to optimal. These results justify the standard practice of hill-climbing to a locally-optimal theory, based on a given set of labeled samples.

1 Introduction

There are many fielded knowledge-based systems, ranging from expert systems and logic programs to production systems and database management systems [Lev84]. Each such system uses its database of general information (a.k.a. its "theory") to produce an answer to each given query; this can correspond to retrieving information from a database (*e.g.*, finding `X` such that "`(makes acme X) & (color X red)`") or to providing a diagnosis or repair, based on a given set of symptoms. Unfortunately, these responses may be incorrect if the underlying theory includes erroneous information. If we observe that some answers are incorrect (*e.g.*, if the patient does not get better, or the proposed repair does not correct the device's faults), we can then ask a human expert to supply the correct answer. *Theory revision* is the process of using such correctly-answered queries to modify the initial theory, to produce a new theory that is more accurate; *i.e.*, which will not make those mistakes again, while remaining correct on the other queries.

Most theory revision algorithms use a set of transformations to hill-climb through successive theories, until reaching a theory whose accuracy is (locally) optimal, based on a set of correctly-answered queries; *cf.*, [Pol85; MB88; Coh90; OM94; WP93; CS90; LDRG94]. This report addresses the obvious question: Is there a better approach, which will directly yield the *globally* optimal theory?

Section 2 first states the theory revision objective more precisely: as finding the theory with the highest accuracy from the space of theories formed by applying a sequence of transformations to a given initial theory; here, each transform involves either adding or deleting either a rule or an antecedent. It also proves that a polynomial number of training "labeled queries" (each a specific query paired with its correct answer) is sufficient; *i.e.*, they provide the information needed to identify a transformation-sequence that will transform the given theory into a new theory whose accuracy is arbitrarily close to optimal, with arbitrarily high probability. Section 3 then addresses the *computational* complexity of the task of finding the optimal (or even near-optimal) revised theory. It first proves that the task of computing the optimal theory within this space of theories is intractable, even in trivial contexts — *e.g.*, even when dealing with propositional Horn theories, or when considering with only atomic queries, or when considering only a bounded number of transformations, etc.[1] We then show that this task cannot even be approximated: *i.e.*, that no efficient algorithm can find a theory whose inaccuracy is even close to (*i.e.*, within a particular small polynomial of) optimum! We also prove that these negative results apply even when we are only generalizing, or only specializing, the initial theory. We also discuss the

*I gratefully acknowledge receiving helpful comments from George Drastal, R. Bharat Rao, Narendra Gupta, Tom Hancock, Sheila McIlraith, Edoardo Amaldi, Dan Roth and Roni Khardon.

[1]Throughout, we will assume that $P \neq NP$ [GJ79], which implies that any NP-hard problem is intractable. This also implies certain approximation claims, presented below.

efficiency of other restricted variants of theory revision, providing sharp boundaries that describe exactly when this task is, versus is not, tractable.

We view these results as sanctioning the standard approach of using a set of transformations to hill-climb to a local optimum, based on a set of samples: The labeled training samples are required to obtain the needed distribution information, and the realization that no tractable algorithm will be able to find the *global* optimum justifies hill-climbing to a local optimum, within the space formed using specified transformations.

We close this section by describing related research.[2]

Related Research: While most learning systems begin with an "empty theory" and attempt to learn a target function (perhaps a decision tree, or a logic program), theory revision processes work by modifying a given initial theory. There are several implemented theory revision systems. Most use essentially the same set of transformations we describe — *e.g.*, AUDREY [WP93], FONTE [MB88], EITHER [OM94] and Δ [LDRG94] all consider adding or deleting antecedents or rules. Our analysis, and results, can easily be applied to many other types of modifications — *e.g.*, specializing or generalizing antecedents [OM94], using "n-of-m rules" [BM93], or merging rules and removing chains paths of rules that produced incorrect results [Coh90; Coh92].[3] While these projects provide empirical evidence of the effectiveness of their specific algorithms, and deal with classification (*i.e.*, determining whether a given element is a member of some target class) rather than general derivation, our work formally addresses the complexities inherent in finding the best theory, for handling arbitrary queries.

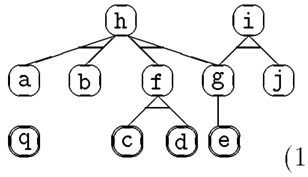
Finally, note that, in some cases, our task can require extracting the best consistent sub-theory from a given inconsistent theory. From this perspective, our work is related to "Knowledge Representation" form of theory revision, *á la* Gärdenfors [Gar88; AGM85], Katsuno and Mendelzon [KM91] and many others. Our work differs by using the notion of expected accuracy to dictate which of the "revisions" is best.

2 Framework

We define a "theory" as a collection of (propositional or first order) Horn clauses, where each clause is a disjunction of literals, at most one of which is positive. Borrowing from [Lev84; DP91], we also view a theory T as a function that maps each query to its proposed answer; hence, T: $\mathcal{Q} \mapsto \mathcal{A}$, where $\mathcal{Q}$ is a (possibly infinite) set of Horn queries, and $\mathcal{A} = \{$ `No`, `Yes` $\}$ is the set of possible answers.[4] Hence, given

T_1 =

```
h :- a, b.
h :- f, g.
i :- g, j.
f :- c, d.
g :- e.
q.  c.  d.  e.
```



(1)

T_1(`h`) = `Yes`, T_1(`i`) = `No` and T_1(`i :- e,j.`) = `Yes`. Of course, different theories can return different answers to a given query: For example, let T_2 be a theory that differs from T_1 only by excluding the "`g :- e`" rule; then T_2(`h`) = `No`.

While the non-atomic queries may seem unusual at first, they are actually quite common. For example, a medical expert system typically collects relevant data $\{$ `f`$_1$`(p)`, …, `f`$_n$`(p)` $\}$ about an individual patient `p`, then determines whether `p` has some specific disease `disease`$_i$; *i.e.*, if $T \cup \{$ `f`$_1$`(p)`, …, `f`$_n$`(p)` $\} \models$ `disease`$_i$`(p)`, where T is the expert system's initial theory that contains general information about diseases, etc. Notice this entailment condition holds iff $T \models \neg$`f`$_1$`(p)` $\vee \ldots \vee \neg$`f`$_n$`(p)` $\vee$ `disease`$_i$`(p)`; *i.e.*, iff the Horn query "`disease`$_i$`(p) :- f`$_1$`(p), …, f`$_n$`(p)`" follows from the initial theory. (See also "entailment queries" [FP93].)

We assume there is a single correct answer to each question, and represent it using the real-world oracle $\mathcal{O} : \mathcal{Q} \mapsto \mathcal{A}$. Here, perhaps, $\mathcal{O}$(`h`) = `No`, meaning that "`h`" should not hold. We say an oracle is *consistent* if it produces the same answers as a Horn theory, over the set of queries $\mathcal{Q}$. *N.b.*, we will sometimes deal with inconsistent oracles; *e.g.*, where $\mathcal{O}$(`a`) = `Yes`, $\mathcal{O}$(`b :- a`) = `Yes`, and $\mathcal{O}$(`b`) = `No`.

Our goal is to find a theory that is as close to $\mathcal{O}(\cdot)$ as possible. To quantify this, we first define the "accuracy function" $a(\cdot, \cdot)$ where $a(\mathrm{T}, q)$ is the accuracy of the answer the theory T returned for the query q:

$$a(\mathrm{T}, q) \quad \stackrel{def}{=} \quad \begin{cases} 1 & \text{if } \mathrm{T}(q) = \mathcal{O}(\,q\,) \\ 0 & \text{otherwise} \end{cases}$$

(Notice $a(\mathrm{T}, \cdot)$ implicitly depends on the oracle $\mathcal{O}(\cdot)$.) Hence, as $\mathcal{O}$(`h`) = `No`, $a(\mathrm{T}_2$, "`h`"$) = 1$ as T_2 provides the correct answer while $a(\mathrm{T}_1$, "`h`"$) = 0$ as T_1 returns the wrong answer.

This $a(\mathrm{T}, \cdot)$ function measures T's accuracy for a single query. In general, our theories must deal with a range of queries. We model this using a stationary, but unknown, probability function $Pr : \mathcal{Q} \mapsto [0, 1]$, where $Pr(q)$ is the probability that the query q will be posed. Given this distribution, we can compute the "expected accuracy" of a theory, T:

$$\mathrm{A}(\,\mathrm{T}\,) \;=\; E[\,a(\mathrm{T}, q)] \;=\; \sum_{q \in \mathcal{Q}} Pr(q) \times a(\mathrm{T}, q)\;.$$

We will consider various sets of possible theories, $\mathcal{T} = \{\mathrm{T}_i\}$, where each such $\mathcal{T}$ contains the set of theories

[2]The technical report [Gre95b] provides a more extensive literature survey, as well as proofs of the theorems.

[3]The companion paper [Gre95a] considers yet other ways of modifying a theory, *viz.*, by rearranging its component rules or antecedents.

[4]To simplify the presentation, the bulk of this paper will deal only with propositional logic; Section 2.3 below describes the extensions needed to deal with predicate calculus.

formed by applying various sequences of transformations to a given initial theory; see Section 2.1 below. Our challenge is to identify the theory $T_{opt} \in \mathcal{T}$ whose expected accuracy is optimal; *i.e.*,

$$\forall\, T \in \mathcal{T}:\ A(\, T_{opt}\,)\ \geq\ A(\, T\,)\,. \qquad (2)$$

There are two challenges to finding such optimal theories. The first is based on the observation that the expected accuracy of a theory depends on the distribution of queries, which means different theories will be optimal for different distributions. While this distribution is not known initially, it can be estimated by observing a set of samples (each a query/answer pair), drawn from that distribution. Section 2.2 below discusses the number of samples required to be confident of obtaining the information needed to identify a good $T^* \in \mathcal{T}$, with high probability.

We are then left with the challenge of computing the best theory, once given this distributional estimate. Section 3 addresses the computational complexity of this process, showing that the task is not just intractable,[5] but it is also not approximatable — *i.e.*, no efficient algorithm can even find a theory whose expected accuracy is even close (in a sense defined below) to the optimal value.

We first close this section by describing the transformations we will use to define the various spaces of theories, then discussing the sample complexity of the implied learning process and finally providing the extensions needed to deal with predicate calculus.

2.1 Standard Transformations

Standard theory revision algorithms implicitly explore the space of possible theories $\Sigma^{\infty}[T_0] = \{\sigma(T_0) \mid \sigma \in \Sigma^{\infty}\}$, which contains the theories formed by applying some sequence of theory-to-theory transformations $\sigma \in \Sigma^{\infty}$ to the given initial theory T_0. Each $\sigma = \tau_1 \circ \tau_2 \circ \ldots \circ \tau_k \in \Sigma^{\infty}$ sequence is formed from $\Sigma = \Sigma_{DR} \cup \Sigma_{AR} \cup \Sigma_{DA} \cup \Sigma_{AA}$, where each $\tau_{DR} \in \Sigma_{DR}$ deletes a rule from the theory, each $\tau_{AR} \in \Sigma_{AR}$ adds a new rule to the theory, each $\tau_{DA} \in \Sigma_{DA}$ deletes an existing antecedent from an existing rule, and each $\tau_{AA} \in \Sigma_{AA}$ adds a new antecedent to an existing rule. In some situations, we will consider "K-bounded sequences"

$$\Sigma^{K} \quad = \quad \{\, \sigma = \tau_1 \circ \tau_2 \circ \ldots \circ \tau_k \mid \tau_i \in \Sigma\ \&\ c(\sigma) \leq K\,\}$$

whose members $\sigma = \tau_1 \circ \tau_2 \circ \ldots \circ \tau_k \in \Sigma^{K}$ are sequences of transformations whose total cost $c(\sigma) = c(\tau_1) + c(\tau_2) + \ldots + c(\tau_k)$ are at most K, where the cost $c(\tau)$ of the transformation τ is the number of symbols added to, or deleted from, T to form $\tau(T)$. In the propositional case, $c(\tau^{AA}) = c(\tau^{DA}) = 1$ for each transformation that either adds or deletes an antecedent; and $c(\tau^{AR}_{\rho}) = c(\tau^{DR}_{\rho}) = |\rho|$ for each add-rule (resp., delete-rule) transformation that adds (resp., deletes) the rule ρ, which has 1 conclusion and $|\rho|-1$ antecedent literals.

As an example, applying the 3-element sequence $\sigma = \tau^{AA}_{g:-e;\ +q} \circ \tau^{AR}_{b:-f} \circ \tau^{DA}_{f:-c,d;\ -c}$ with total cost $c(\sigma) = c(\tau^{AA}_{g:-e;\ +q}) + c(\tau^{AR}_{b:-f}) + c(\tau^{DA}_{f:-c,d;\ -c}) = 1 + 2 + 1 = 4$, will transform T_1 into $\sigma(T_1) = \tau^{AA}_{g:-e;\ +q}(\,\tau^{AR}_{b:-f}(\,\tau^{DA}_{f:-c,d;\ -c}(\,T_1\,)\,)\,)$ which is a theory with 8 clauses that differs from T_1 by including the clause "`g:-e,q`" rather than "`g:-e`", including the clause "`f:-d`" rather than "`f:-c,d`", and by including an extra clause "`b:-f`".

Finally, we will also consider various other restricted spaces of transformation-sequences, which are formed from specified types of transformations; *e.g.*,

$$\Sigma^{-R,\,+A} \;=\; \{\sigma = \tau_1 \circ \tau_2 \circ \ldots \circ \tau_k \mid \tau_i \in \Sigma_{DR} \cup \Sigma_{AA}\,\}$$
$$\Sigma^{+R,\,-A} \;=\; \{\sigma = \tau_1 \circ \tau_2 \circ \ldots \circ \tau_k \mid \tau_i \in \Sigma_{AR} \cup \Sigma_{DA}\,\}$$

correspond to (unbounded) transformation sequences that produce more specific (resp., more general) theories; as well as the bounded variants; *e.g.*,$\Sigma^{-R,\,+A\ (K)} = \{\sigma = \tau_1 \circ \ldots \circ \tau_k \mid \tau_i \in \Sigma_{DR} \cup \Sigma_{AA}\ \&\ c(\sigma) \leq K\}$. Note that the earlier $\Sigma^{\infty} = \Sigma^{+R,-R,+A,-A}$ and $\Sigma^{K} = \Sigma^{+R,-R,+A,-A\,(K)}$.

2.2 Sample Complexity

We can use following standard Computational Learning Theory theorem to bound the number of samples required to obtain the information needed to identify a good $T^* \in \mathcal{T}$ with high probability; showing in particular how this depends on the space of theories $\mathcal{T}$ being considered:

Theorem 1 (from [Vap82, Theorem 6.2]) *Given a class of theories $\mathcal{T}$ and $\epsilon, \delta > 0$, let $T_* \in \mathcal{T}$ be the theory with the largest empirical accuracy after*

$$M_{upper}(\mathcal{T}, \epsilon, \delta) \quad = \quad \left\lceil \tfrac{2}{\epsilon^2} \ln\left(\tfrac{|\mathcal{T}|}{\delta}\right)\right\rceil$$

samples (each a labeled query), drawn from the stationary distribution, $Pr(\cdot)$. Then, with probability at least $1-\delta$, the expected accuracy of T_* will be within ϵ of the optimal theory in $\mathcal{T}$; i.e., $Pr[\,A(\,T_*\,) \geq A(\,T_{opt}\,) - \epsilon\,] \geq 1-\delta$, using the T_{opt} from Equation 2.

This means a polynomial number of samples is sufficient to identify an ϵ-good theory from $\mathcal{T}$ with probability at least $1-\delta$, whenever $\ln(|\mathcal{T}|)$ is polynomial in the relevant parameters. Notice this is true for most of the classes of theories being considered; *e.g.*, as $\Sigma^{-R}(T)$ is the power-set of the rules in T, $|\Sigma^{-R}(T)| = 2^{|\mathrm{Rules}(T)|}$, which means $|\ln(\Sigma^{-R}(T))| = |\mathrm{Rules}(T)| = O(|T|)$, which clearly is polynomial in the size of the initial theory. This "$\ln(|\mathcal{T}|) = \mathrm{poly}(|T|)$" claim is slightly problematic for transformations that can add symbols, notably for Σ^{+R} and Σ^{+A}. But even here, the sample complexity remains polynomial *in the size of the revised theory*, which effectively means again that sample-efficient learning remains possible; *cf.*, "nonuniform" pac-learning [BI88]

[5] As $a(T, q)$ requires computing $T(q)$, which can require proving an arbitrary theorem, this computation alone can be computationally intractable, if not undecidable. Our results show that the task of finding the optimal theory is intractable *even given a poly-time oracle for these arbitrary derivations.* Of course, as we are considering only Horn theories, these computations are guaranteed to be poly-time in the propositional case [BCH90].

2.3 Dealing with Predicate Calculus

To handle predicate calculus expressions, we have to consider answers of the form $\{\texttt{Yes}[\ \{X_i/\mathtt{v}_i\}\ ]\}$, where the expression within each $\texttt{Yes}[\cdot]$ is a binding list of the free variables, corresponding to a single answer to the query. For example, given the theory[6]

$$\mathrm{T}_{pc} = \left\{ \begin{array}{ll} \texttt{tall(john).} & \texttt{short(fred).} \\ \texttt{rich(john).} & \texttt{rich(fred).} \\ \multicolumn{2}{l}{\texttt{eligible(X) :- tall(X), rich(X).}} \end{array} \right\}$$

the query `short(Y)` will return $\mathrm{T}_{pc}(\texttt{short(Y)}) = \{\texttt{Yes}[\{Y/\texttt{fred}\}]\}$, the query `rich(Z)` will return the pair of answers $\mathrm{T}_{pc}(\texttt{rich(Z)}) = \{\texttt{Yes}[\{Z/\texttt{john}\}], \texttt{Yes}[\{Z/\texttt{fred}\}]\}$, and $\mathrm{T}_{pc}(\texttt{eligible(A)}) = \{\texttt{Yes}[\{A/\texttt{john}\}]\}$. As $\mathcal{O}(\cdot)$ and $\mathrm{T}(\cdot)$ each returns a *set* of answers to each query, we therefore define T's accuracy score as $a(\mathrm{T}, \sigma) = \frac{|\mathcal{O}(\sigma)\cap T(\sigma)|}{|\mathcal{O}(\sigma)\cup T(\sigma)|} \in [0,1]$. All of the theorems in this paper hold even when considering only non-recursive Datalog (*i.e.*, "function-free") theories.

3 Computational Complexity

Our basic challenge is to produce a theory T_{opt} whose accuracy is as large as possible. The previous section supplied the number of samples needed to guarantee, with high probability, that the expected accuracy of the theory whose empirical accuracy is largest, T_*, will be within ϵ of the expected accuracy of this T_{opt}. This section discusses the computational challenge of determining this T_*, given this distributional estimate. We show first that this task is tractable in degenerate trivial situations: when considering (1) only *atomic* queries posed to a (2) *propositional* theory and being allowed (3) an *arbitrarily large number of modifications* to the initial theory, to produce (4) a *perfect* theory (*i.e.*, one that returns the correct answer to *every* query). This task becomes intractable, however, if we remove (essentially) any of these restrictions: *e.g.*, if we seek optimal (rather than only seeking "perfect") propositional theories and are allowed to pose Horn queries, or if we consider *predicate calculus* theories. It also remains intractable even if we restrict the *number* of modifications allowed, which implies that the task of determining the *smallest* number of modifications required to find a perfect theory is intractable. We next show that these tasks are not just intractable but worse, they are not even approximatable, except in the most trivial of situations.

We also consider two special subtasks by restricting the allowed types of transformations, to consider revision processes that only specialize (respectively, only generalize) the initial theory. We show that these tasks, also, are intractable and non-approximatable in essentially all situations; *i.e.*, except when all four of the above conditions hold.[7] Figures 1 and 2 summarize the various cases.

[6] Following PROLOG's conventions, we will capitalize each variable, as in the "X" above.

[7] Actually, there is one other tractable case in the generalization situation; see Figure 1.

3.1 Basic Complexity Results

To formally state the problem:

Definition 1 ($\mathrm{TR}[\Sigma^\dagger]$ Decision Problem)
INSTANCE:
- *Initial theory* T;
- *Labeled training sample* $S = \{\langle q_i, \mathcal{O}(q_i)\rangle\}$ *containing a set of Horn queries and the correct answers; and*
- *Probability value* $p \in [0,1]$.

QUESTION: *Is there a theory* $\mathrm{T}' \in \Sigma^\dagger[\mathrm{T}]$ *such that*
$$A_S(\mathrm{T}') = \tfrac{1}{|S|}\sum_{\langle q_i, \mathcal{O}(q_i)\rangle \in S} a(\mathrm{T}', q_i) \ \geq\ p\,?$$

The $\Sigma^\dagger[\cdot]$ function maps a theory to a set of candidate revised theories; here, we will consider various $\Sigma^{\pm R, \pm A}$ transformation sets. To simplify our notation, we will write $\mathrm{A}(\mathrm{T})$ for $\mathrm{A}_S(\mathrm{T})$.

We will also consider the following special cases: $\mathrm{TR}_{Perf}[\Sigma^\dagger]$ requires that $p = 1$ (*i.e.*, seeking perfect theories, rather than "optimal" theories $\mathrm{TR}_{(Opt)}[\Sigma^\dagger]$); $\mathrm{TR}_{Prop}[\Sigma^\dagger]$ deals with propositional logic (rather than predicate calculus, $\mathrm{TR}_{(PredCal)}[\Sigma^\dagger]$); and $\mathrm{TR}_{Atom}[\Sigma^\dagger]$ deals with only atomic queries (as opposed to Horn queries, $\mathrm{TR}_{(Horn)}[\Sigma^\dagger]$). We will also use $\mathrm{TR}_{Disj}[\Sigma^\dagger]$ to refer to the task when the queries can be arbitrary disjunctions, which need not be Horn. (While the other subscripts are *restrictions* on $\mathrm{TR}[\Sigma^\dagger]$, this *Disj* case is more permissive.) We will also combine subscripts, with the obvious meanings. When $\mathrm{TR}_\chi[\Sigma^\dagger]$ is a special case of $\mathrm{TR}_\psi[\Sigma^\dagger]$, finding that $\mathrm{TR}_\chi[\Sigma^\dagger]$ is hard immediately implies that $\mathrm{TR}_\psi[\Sigma^\dagger]$ is hard. Similarly, seeing that $\mathrm{TR}_\psi[\Sigma^\dagger]$ is easy immediately implies that each special case of $\mathrm{TR}_\psi[\Sigma^\dagger]$ is easy. As a final note: all of the hardness results presented in this paper hold even if we only consider "3-CNF Horn theories" — *i.e.*, rules whose antecedents contain at most 2 literals.

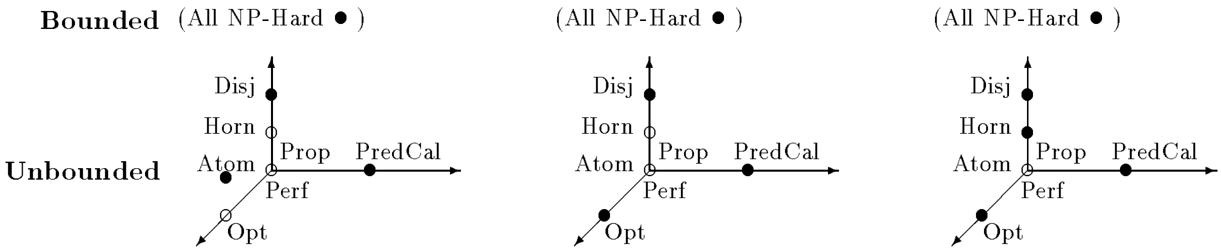
Here, it is easy to find the optimal theory in certain degenerate cases, where either the individual queries can be decoupled (*e.g.*, when using atomic propositional queries) or when our actions are forced (*e.g.*, when seeking perfect propositional theories): just throw away the original theory, then add in propositions corresponding to the "`Yes`-labeled queries". In every other case, however, the task is intractable:

Theorem 2 *The* $\mathrm{TR}_{Prop,Atom,(Opt)}[\Sigma^\infty]$ *and* $\mathrm{TR}_{Prop,(Horn),Perf}[\Sigma^\infty]$ *decision problems are easy; each other problem is NP-hard.*

(This information is summarized in lower left "Unbounded, Arbitrary" graph of Figure 1.)

The above theorem describes the complexity of computing the best theory when we are allowed to use an *arbitrarily expensive* sequence of transformations. We get an even stronger negative results if we restrict the "expense" of the transformation sequence:

Theorem 3
For some $K \in \mathcal{Z}^+$, *the* $\mathrm{TR}_{Prop,Atom,Perf}[\Sigma^K]$ *decision problem is NP-hard. This is true even if we consider only labeled queries produced by a consistent oracle (i.e.,*



Arbitrary (Σ^K) **Generalization** ($\mathcal{G}$) **Specialization** ($\mathcal{S}$)

○ = Easy to solve; ● = NP-hard
Bounded = "$K < \infty$"; **Unbounded** = "$K = \infty$"

$$\begin{array}{lcl} \textbf{Arbitrary } \Sigma^K & = & \Sigma^{+R,\ -A,\ -R,\ +A\ (K)} \\ \textbf{Generalization } \mathcal{G} & \in & \{\ \Sigma^{+R,\ -A\ (K)},\ \Sigma^{+R\ (K)},\ \Sigma^{-A\ (K)}\ \} \\ \textbf{Specialization } \mathcal{S} & \in & \{\ \Sigma^{-R,\ +A\ (K)},\ \Sigma^{-R\ (K)},\ \Sigma^{+A\ (K)}\ \} \end{array}$$

Any task that "projects" down to an NP-hard task, along any axis, is NP-hard. Here, this means all of the "cross terms" are NP-hard. (For example $\text{ThRev}_{PredCal,Horn,Perf}[\Sigma^\infty]$ is NP-hard, as its projection to the "Prop–PredCal × Perf–Opt" plane, $\text{ThRev}_{PredCal,Atom,Perf}[\Sigma^\infty]$ is NP-hard.) The $\text{ThRev}_{Prop,Horn,Opt}[\Sigma^\infty]$ case is shown explicitly as each of its projections is easy; the figures omit all other cross-terms.

Figure 1: Tractability of Theory Revision Tasks

even when there is a Horn theory that correctly labels all of the queries).

The observation that determining such "K-step perfect theories" is NP-hard leads immediately to:

Corollary 3.1 *It is NP-hard to compute the minimal-cost transformation sequence required to produce a perfect theory (i.e., to compute the smallest K for which there is a $\mathrm{T}_{perfect} \in \Sigma^K[T]$ such that $A(\mathrm{T}_{perfect}) = 1$), even in the propositional case when considering only atomic queries. It is also NP-hard to compute the "minimal-length" transformation, where the length of the transformation sequence $\tau_1 \circ \tau_2 \circ \ldots \circ \tau_k$ is simply k — i.e., when each transformation has "unit cost".*

This negative result shows the intractability of the obvious proposal of using a breath-first transversal of the space of all possible theory revisions: First test the initial theory T_0 against the labeled queries, and return T_0 if it is 100% correct. If not, then consider all theories formed by applying a single (unit-cost) transformation, and return any perfect $\mathrm{T}_1 \in \Sigma^1[\mathrm{T}_0]$; and if not, consider all theories in $\Sigma^2[\mathrm{T}_0]$ (formed by applying sequences of transformations with cost at most two), and return any perfect $\mathrm{T}_2 \in \Sigma^2[\mathrm{T}_0]$; and so forth.

3.2 Approximatability

Many decision problems correspond immediately to optimization problems; for example, the MinGraphColor decision problem (given a graph $G = \langle N, E\rangle$ and a positive integer K, can each node be labeled by one of K colors in such a way that no edge connects two nodes of the same color; see [GJ79, p191(Chromatic Number)]) corresponds to the obvious minimization problem: Find the minimal coloring of the given graph G. We can similarly view the $\mathrm{TR}_\chi[\Sigma^\dagger]$ decision problem as either the maximization problem: "Find the $\mathrm{T}' \in \Sigma^\dagger[\mathrm{T}]$ whose accuracy is maximal" or the minimization problem: "Find the $\mathrm{T}' \in \Sigma^\dagger[\mathrm{T}]$ whose inaccuracy is minimal", where a theory's inaccuracy is obviously $\mathrm{InA}(\mathrm{T}) = 1 - \mathrm{A}(\mathrm{T})$. (While the maximally accurate theory is also minimally inaccurate, these two formulations can lead to different approximatability results.) For notation, let "$\mathrm{MaxTR}_\chi[\Sigma^\dagger]$" (resp., "$\mathrm{MinTR}_\chi[\Sigma^\dagger]$") refer to the maximization (resp., minimization) problem.

Now consider any algorithm B that, given any $\mathrm{MinTR}_\chi[\Sigma^\dagger]$ instance $x = \langle T, S\rangle$ with initial theory T and labeled training sample S, computes a syntactically legal, but not necessarily optimal, revision $B(x) \in \Sigma^\dagger[\mathrm{T}]$. Then B's "performance ratio for the instance x" is defined as

$$MinPerf[\mathrm{MinTR}_\chi[\Sigma^\dagger]](B, x) = \begin{cases} \frac{InA(B(x))}{InA(opt(x))} & \text{if } \mathrm{InA}(opt(x)) \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $opt(x) = opt_{MinTR_\chi(\Sigma^\dagger)}(x)$ is the optimal solution for this instance; *i.e.*, $opt(x)$ is the theory $\mathrm{T}_{opt} \in \Sigma^\dagger[\mathrm{T}]$ with minimal inaccuracy over S.

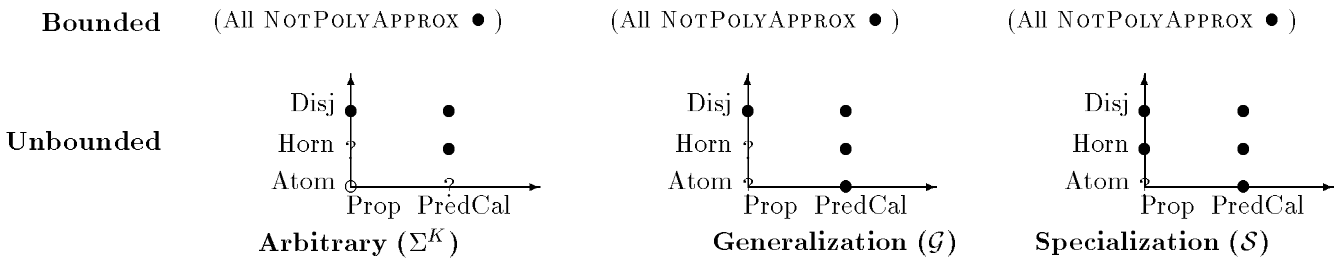
We say a function $g(\cdot)$ "bounds B's performance ratio (over $\mathrm{MinTR}_\chi[\Sigma^\dagger]$)" iff

$$\forall x \in \mathrm{MinTR}_\chi[\Sigma^\dagger], \quad MinPerf[\mathrm{MinTR}_\chi[\Sigma^\dagger]](B, x) \le g(|x|)$$

where $|x|$ is the size of the instance $x = \langle \mathrm{T}, S\rangle$, which we define to be the number of symbols in T plus the number of symbols used in S. Intuitively, this $g(\cdot)$ function indicates how closely the B algorithm comes to returning the best answer for x, over all $\mathrm{MinTR}_\chi[\Sigma^\dagger]$ instances x.

Now let $Poly(\mathrm{MinTR}_\chi[\Sigma^\dagger])$ be the collection of all polytime algorithms that return legal answers to $\mathrm{MinTR}_\chi[\Sigma^\dagger]$ instances. It is natural to ask for the algorithm in $Poly(\mathrm{MinTR}_\chi[\Sigma^\dagger])$ with the best performance ratio; this would indicate how close we can come to the optimal solution, using only a feasible computational time. For example, if this function was the constant 1 for $\mathrm{MinTR}_{Prop}[\Sigma^\infty]$, then a poly-time algorithm could produce the optimal solution to any $\mathrm{MinTR}_{Prop}[\Sigma^\infty]$ instance; as $\mathrm{TR}_{Prop}[\Sigma^\infty]$ is NP-complete,[8] this would

[8] While Theorem 2 only proves $\mathrm{TR}_{Prop}[\Sigma^\infty]$ to be NP-



(● = NotPolyApprox; ○ = Easy (as poly-time decision); ? = unknown approximatability class)

Figure 2: Approximatability of Theory Revision Tasks

mean $P = NP$, which is why we do not expect to obtain this result. Or if this bound was some constant $c(x) = c \in \Re$, then we could efficiently obtain a solution within a factor of c of optimal, which may be good enough for some applications.[9]

However, not all problems can be so approximated. Following [CP91; Kan92], we define

Definition 2 *A minimization problem* MinP *is* NotPolyApprox *if there is a* $\gamma \in \Re^+$ *such that*

$$\forall B \in \text{Poly}(\,\text{MinP}\,),\ \exists x \in \text{MinP},\quad \textit{MinPerf}[\text{MinP}](\,B, x\,) \ \geq\ |x|^\gamma.$$

Lund and Yannakakis [LY93] prove that the "MinGraphColor minimization problem" is NotPolyApprox. We can use that result to prove:

Theorem 4 *Unless* $P = NP$, *each of* $\text{MinTR}_{Prop,Disj}[\Sigma^\infty]$, $\text{MinTR}_{(PredCal),(Horn)}[\Sigma^\infty]$ *and* $\text{MinTR}_{Prop,Atom}[\Sigma^K]$ *is* NotPolyApprox.

While these results may at first seem trivial, given that it is NP-hard to determine if a perfect theory exists, notice from Equation 3 that $\textit{MinPerf}[\text{MinTR}[\Sigma^\infty]](\cdot)$ essentially ignores such perfect theories. Note also that this result holds in the context based on an "inconsistent" oracle; in such situations, no theory can be perfect.

As $|x|$ can get arbitrary large, this result means that these $\text{MinTR}_\chi[\Sigma^\dagger]$ tasks cannot be approximated by any constant, nor even by any logarithmic factor nor any sufficiently small polynomial, etc.

3.3 Special Cases

If the theory is too general (*i.e.*, returns `Yes` too often), then we may want to consider "specializing" it by applying only the "delete rule" and "add antecedent" transformations. In particular, recall that $\Sigma^{+A,-R}[\text{T}]$ is the set of theories obtained using *an arbitrary number* of such transformations, and $\Sigma^{-R}[\text{T}]$ (resp., $\Sigma^{+A}[\text{T}]$), is the set of theories obtained by applying an arbitrary number of "delete rule" (respectively, "add antecedent") transformations. Similarly, if the theory is too specific (*i.e.*, returns `No` too often), then we may want to consider "generalizing" it by applying only the "add rule" and "delete antecedent" transformations; here, we consider $\Sigma^{+R,-A}[\text{T}]$, $\Sigma^{+R}[\text{T}]$ and $\Sigma^{-A}[\text{T}]$, which are the set of theories obtained by applying an arbitrary number of such transformations.

Even using only these transformations, almost all of these tasks remain intractable:

Theorem 5 *It is easy to solve*
$\text{TR}_{Prop,Perf}[\mathcal{G}]$ *for* $\mathcal{G} \in \{\Sigma^{+R,-A}, \Sigma^{+R}, \Sigma^{-A}\}$,
$\text{TR}_{Prop,Atom,Perf}[\mathcal{S}]$ *for* $\mathcal{S} \in \{\Sigma^{-R,+A}, \Sigma^{-R}, \Sigma^{+A}\}$.
However, every other situation*, formed by any other combination of restrictions (read "subscripts") is NP-hard. (See middle and right of Figure 1.)*

Worse,

Theorem 6 *Unless* $P = NP$, *each of the following is* NotPolyApprox*:*

- $\text{MinTR}_{(PredCal),Atom}[\mathcal{S}]$, $\text{MinTR}_{Prop,(Horn)}[\mathcal{S}]$ *for* $\mathcal{S} \in \{\Sigma^{-R,+A},\ \Sigma^{-R},\ \Sigma^{+A}\ \}$
- $\text{MinTR}_{(PredCal),Atom}[\mathcal{G}]$, $\text{MinTR}_{Prop,Disj}[\mathcal{G}]$ *for* $\mathcal{G} \in \{\Sigma^{+R,-A},\ \Sigma^{+R},\ \Sigma^{-A}\ \}$.
- $\text{MinTR}_{Prop,Atom}[\Sigma^\dagger]$ *for* $\Sigma^\dagger \in \left\{ \begin{array}{l} \Sigma^{+A,-R(K)},\ \Sigma^{-R(K)},\ \Sigma^{+A(K)} \\ \Sigma^{-A,+R(K)},\ \Sigma^{+R(K)},\ \Sigma^{-A(K)} \end{array} \right\}$

(See middle and right of Figure 2.)

In each of these cases, however, there is a trivial polynomial-time algorithm that can produce a theory whose *accuracy* (*n.b.*, not inaccuracy) is within a factor of 2 of optimal. That is, using the ratio of an algorithm's *accuracy* to the optimal value,

$$\textit{MaxPerf}[\text{MaxTR}_\chi[\Sigma^\dagger]](\,B, x\,) \quad = \quad \frac{A(\,opt(x)\,)}{A(\,B(x)\,)}$$

Theorem 7
For $\Psi \in \{\Sigma^{-R,+A}, \Sigma^{-R}, \Sigma^{+A}, \Sigma^{+R,-A}, \Sigma^{+R}, \Sigma^{-A}\}$,
$\exists B \in \text{Poly}(\,\text{MaxTR}[\Psi]\,)$,
$\textit{MaxPerf}[\text{MaxTR}[\Psi]](\,B, x\,) \ \leq\ 2$

The companion paper [Gre95a] considers other related cases, including the above special cases in the context where our underlying theories can use the `not(·)` operator to return `Yes` if the specified goal cannot be proven; *i.e.*, using Negation-as-Failure [Cla78]. It also considers the effect of re-ordering the rules and the antecedents, in the context where such shufflings can affect the answers returned. In most of these cases, we show that the corresponding maximization problem is not approximatable within a particular polynomial.

---

hard, this problem is clearly in NP.

[9]There are such constants for some other NP-hard minimization problems. For example, there is a polynomial-time algorithm that computes a solution whose cost is within a factor of 1.5 for any TravelingSalesman-with-Triangle_Equality problem; see [GJ79, Theorem 6.5].

(The extended [Gre95b] explains the asymmetry between $\mathrm{TR}_{Prop,Perf}[\Sigma^{-R}]$ versus $\mathrm{TR}_{Prop,Perf}[\Sigma^{+R}]$, and discusses how these results relate to both inductive logic programming, and to default theories.)

4 Conclusion

A knowledge-based system can produce incorrect answers to queries if its underlying theory is faulty. A "theory revision" system attempts to transform a given theory into a related one that is as accurate as possible, using a given set of correctly-answered "training queries". This report describes both the sample and computational complexity of this task. It first provides the number of samples required to obtain the statistics needed to identify a theory (from within a class of theories defined by applying various standard transformations to a given initial theory) whose accuracy will be within ϵ of the optimal theory in this class, with probability at least $1 - \delta$. It then shows that, in general, the task of computing this globally optimal theory is intractable — and worse, that no polynomial time algorithm can be guaranteed to find a solution that is even close to optimal (given the standard $P \neq NP$ assumption). We also present special cases of these tasks, which pin-point exactly when the task becomes tractable.

References

[AGM85] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–30, 1985.

[BCH90] E. Boros, Y. Crama, and P.L. Hammer. Polynomial-time inference of all valid implications for horn and related formulae. *Annals of Mathematics and Artificial Intelligence*, 1:21–32, 1990.

[BI88] G. Benedek and A. Itai. Nonuniform learnability. In *Proceedings ICALP-88*, pages 82–92, 1988.

[BM93] Paul T. Baffes and Raymond J. Mooney. Symbolic revision of theories with M-of-N rules. In *Proceedings of IJCAI-93*, August 1993.

[Cla78] K. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, New York, 1978.

[Coh90] William W. Cohen. Learning from textbook knowledge: A case study. In *Proceeding of AAAI-90*, 1990.

[Coh92] William W. Cohen. Abductive explanation-based learning: A solution to the multiple inconsistent explanation problems. *Machine Learning*, 8(2):167–219, March 1992.

[CP91] P. Crescenzi and A. Panconesi. Completeness in approximation classes. *Information and Computation*, 93(2):241–62, 1991.

[CS90] Susan Craw and Derek Sleeman. Automating the refinement of knowledge-based systems. In L.C. Aiello, editor, *Proceedings of ECAI 90*. Pitman, 1990.

[DP91] Jon Doyle and Ramesh Patil. Two theses of knowledge representation: Language restrictions, taxonomic classification, and the utility of representation services. *Artificial Intelligence*, 48(3), 1991.

[FP93] Michael Frazier and Leonard Pitt. Learning from entailment: An application to propositional horn sentences. In *Proceedings of IML-93*, pages 120–27. Morgan Kaufmann, 1993.

[Gar88] Peter Gardenfors. *Knowledge in Flux: Modeling the Dynamics of the Epistemic States*. Bradford Book, MIT Press, Cambridge, MA, 1988.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[Gre95a] Russell Greiner. The challenge of revising impure theories. In *Proceedings of the Twelfth International Machine Learning Conference*, 1995.

[Gre95b] Russell Greiner. The complexity of theory revision. Technical Report SCR-95-TR-539, Siemens Corporate Research, 1995. `ftp://scr.siemens.com/pub/learning/Papers/greiner/comp-tr.ps`.

[Kan92] Viggo Kann. *On the Approximability of NP-Complete Optimization Problems*. PhD thesis, Royal Institute of Technology, Stockholm, 1992.

[KM91] Hirofumi Katsuno and Alberto Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings of KR-91*, pages 387–94, Boston, April 1991.

[LDRG94] Pat Langley, George Drastal, R. Bharat Rao, and Russell Greiner. Theory revision in fault hierarchies. In *Proceedings of The Fifth International Workshop on Principles of Diagnosis (DX-94)*, New Paltz, NY, 1994.

[Lev84] Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23:155–212, 1984.

[LY93] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. In *Proceeding of Twenty-fifth Annual ACM Symposium on Theory of Computation (STOC-93)*, pages 286–93, 1993.

[MB88] S. Muggleton and W. Buntine. Machine invention of first order predicates by inverting resolution. In *Proceedings of IML-88*, pages 339–51. Morgan Kaufmann, 1988.

[OM94] Dirk Ourston and Raymond J. Mooney. Theory refinement combining analytical and empirical methods. *Artificial Intelligence*, 66(2):273–310, 1994.

[Pol85] P.G. Politakis. *Empirical Analysis for Expert Systems*. Pitman Research Notes in Artificial Intelligence, 1985.

[Vap82] V.N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982.

[WP93] James Wogulis and Michael J. Pazzani. A methodology for evaluating theory revision systems: Results with Audrey II. In *Proceedings of IJCAI-93*, pages 1128–1134, 1993.