Learning an Optimally Accurate Representation System

Russell Greiner[1] and Dale Schuurmans[2]

[1] Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540-6632
[2] Dept. of Computer Science, University of Toronto, Toronto, ON M5S 1A4, Canada

Abstract. A default theory can sanction different, mutually incompatible, answers to certain queries. We can identify each such theory with a set of related credulous theories, each of which produces but a single response to each query, by imposing a total ordering on the defaults. Our goal is to identify the credulous theory with optimal "expected accuracy" averaged over the natural distribution of queries in the domain. There are two obvious complications: First, the expected accuracy of a theory depends on the query distribution, which is usually not known. Second, the task of identifying the optimal theory, even given that distribution information, is intractable. This paper presents a method, OptAcc, that side-steps these problems by using a set of samples to estimate the unknown distribution, and by hill-climbing to a local optimum. In particular, given any error and confidence parameters $\epsilon, \delta > 0$, OptAcc produces a theory whose expected accuracy is, with probability at least $1 - \delta$, within ϵ of a local optimum.

1 Introduction

A "representation system" R is a program that produces an answer to each given query. We of course prefer "accurate" answers — i.e., answers that correspond correctly to the world. As obvious examples, we prefer that our R returns the answer "4" to the query "find x such that $2+2=x$", produces the appropriate bid for each hand in bridge, finds the correct diagnosis from a given set of patient symptoms, and so forth. We define R's "expected accuracy" as the percentage of answers that it produces that are correct, averaged over the distribution of queries posed. Our goal is to find the representation system with the largest possible expected accuracy.

Most representation systems base their answers on their store of factual information. When this body of accepted information is insufficient to entail an answer to some queries, many of these systems will consider augmenting this initial information with some new hypothesis (or conjecture or default) that is plausible but not necessarily true; each particular collection of facts and hypotheses is a "default theory" [Rei87]. Unfortunately, there can often be more than one such hypothesis, and these hypotheses (and hence the conclusions they respectively entail) may not be compatible; consider for example the Nixon diamond [Rei87, p155]:

> By default, Quakers tend to be pacifists, while Republicans tend to be non-pacifists. Given that Nixon is both a Quaker and a Republican, should we believe that he is, or is not, a pacifist?

This is called the "multiple extension problem" in the knowledge representation community, and corresponds to the "bias" and "multiple explanation" problems in machine learning, and the "reference class problem" in statistics. In each, it has produced a great deal of attention and debate; *cf.*, [Rei87, Mor87] [Mit80, RG87, Hau88], [Kyb82, Lou88].

In general, an effective representation system will return a single (and we hope, correct) answer to each query, rather than remain silent or propose a set of incompatible answers. We therefore focus on a credulous theories, here formed by embellishing a standard default theory with an ordering on its defaults [vA90, Bre89], with the understanding that only the most preferred default(s) will be used to reach a unique answer to each query; see Section 2.[3] As a theory that produces the correct response for one query may be incorrect for other queries, it is not obvious which of the different credulous theories is best.

We of course prefer theories that are likely to be correct, over the natural distribution of queries encountered in the domain. This leads us to define the best theory as the one whose "expected accuracy", over this distribution of queries, is optimal. Section 2 defines this accuracy criterion more precisely. It also shows that the optimally accurate ordering depends on the the distribution of queries; i.e., one R_1 may be optimal for one distribution, whereas another R_2 may be optimal for another. Unfortunately, this distribution information is usually not known *a priori*. Moreover, the task of identifying the optimal ordering, even given that distribution information, is generally intractable. Section 3 develops a learning method that side-steps these two problems by *(i)* using a set of query/answer pairs to estimate the unknown distribution; and *(ii)* by hill-climbing to a local optimum. In particular, it describes the OptAcc algorithm that, given error and confidence parameters $\epsilon, \delta > 0$, returns an ordering of the hypotheses whose expected accuracy is, with probability at least $1-\delta$, within ϵ of a local optimum. Section 4 then discusses several extensions to both our framework and this algorithm. We close this section by describing other research that is related to our work.

Related Research: Our underlying task, of producing a theory that is as correct as possible, is the *sine qua non* of essentially all research on inductive learning; *cf.*, [MCM83, HV88, Hin89]. While many of these systems learn descriptions based on bit vectors or simple hierarchies, our work deals in the context of propositions; here too there is a history of results, dating back (at least) to Shapiro [Sha83], and including FOIL [Qui90] and the body of work on inductive logic programming [MB88]. While much of that research deals with *monotonic* (usually propositional or first order logic) theories and discusses ways of *extending* such theories, producing new theories that can return additional answers, we

[3] Subsection 4.4 presents one way of allowing a "credulous" system to remain skeptical in certain situations.

instead deal with *default* theories, which distinguish between hard, unquestionable facts versus plausible but possibly erroneous defaults, and describe a way of *restricting* a given (default) theory, to produce fewer answers; here, seeking a "weakened" variant that will produce only the correct answer to each question, and not the incorrect one. Many other bodies of research also seek weakened theories (i.e., theories which admit fewer conclusions), albeit in the framework of standard monotonic theories. (1) One branch of explanation-based learning (EBL) research seeks the appropriate "specialization" (read "weakenings") of a given theory [FD89, OM90, Paz88, Coh92]; however, (*i*) the underlying performance task [BMSJ78] for the EBL systems is classification (i.e., determining whether a given element is, or is not, a member of some target class) rather than general derivation; and (*ii*) each uses negation-as-failure [Cla78] (a hard-wired form of non-monotonicity) to classify negatively any sample that cannot be proved to be in the class. By contrast, our work can accommodate general queries, and deals with general default theories. (2) If we coalesce our facts and defaults, we have in essence an inconsistent (monotonic) theory, from which we want to extract the best consistent sub-theory. From this perspective, our work is also related to one form of "theory revision", *á la* [Gar88, AGM85] and many others. Two major distinctions are (*i*) our work explicitly constrains the set of propositions that can be affected (*viz.*, only hypotheses can be deleted); and (*ii*) we use an explicit notion of expected accuracy to dictate which of the possible revisions (read "weakenings") to use. (3) The work on "approximation" [BE89, SK91, DE92, GS92] also seeks good weakenings. Its goal however is an *efficient* encoding; by contrast, we are seeking an *accurate* representation. Finally, the motivation underlying our work is similar to the research in [Sha89] and elsewhere, which also uses probabilistic information to identify the best default theory. Our research differs by using statistical sampling techniques to obtain estimates of the required distribution, and by coping with the computational complexity inherent in this identification process.

2 Framework

This section first provides the general framework for our analysis, then describes the class of representation systems we will use.

2.1 General Analytic Framework

Following [Lev84] and [DP91], we view a representation system R as a function that maps each query to its proposed answer; hence, R: $\mathcal{Q} \mapsto \mathcal{A}$, where $\mathcal{Q}$ is a (possibly infinite) set of queries, and $\mathcal{A}$ is the set of possible answers. Here, we focus on $\mathcal{A} = \{$ `No`, `IDK`, `Yes`$[?x_i \mapsto \mathtt{V}_i]\}$, where `IDK` stands for the non-categorical answer "I Don't Know", and the mapping within the `Yes`'s brackets is a binding list of free variables.[4] Hence, perhaps $\mathrm{R}_1($"$2+2 = ?x$"$) =$

[4] Section 4 presents several extensions to this framework. Also, by convention, the name of each variable will start with a "?", as in "$?x$" here.

$\texttt{Yes}[?x \mapsto \texttt{4}]$, $\mathrm{R}_1(\text{“}2+2 = 19\text{”}) = \texttt{No}$, and $\mathrm{R}_1(\text{“P = NP”}) = \texttt{IDK}$. Of course, different representation systems can return different answers to a given query (e.g., $\mathrm{R}_1(\text{“}\texttt{Pacifist(Nixon)}\text{”}) = \texttt{Yes}[]$ and $\mathrm{R}_2(\text{“}\texttt{Pacifist(Nixon)}\text{”}) = \texttt{No}$) and they can be incorrect; e.g., $\mathrm{R}_1(\text{“}\texttt{Pacifist(Ghandi)}\text{”}) = \texttt{No}$, or $\mathrm{R}_2(\text{“}2+2 = 7\text{”}) = \texttt{Yes}[]$, etc. We will assume that there is a single correct, categorical answer to each question; and represent it using the $\mathcal{O}_{qa} : \mathcal{Q} \mapsto \mathcal{A}$ real-world oracle. (This oracle can be the "real world" that provides the real answers to queries posed. Notice $\mathcal{O}_{qa}[\,\cdot\,]$ is categorical, meaning it will never return "IDK".)

In general, we will consider a given set of possible representation systems, $\mathcal{R} = \{\mathrm{R}_i\}$; below each $\mathrm{R}_i \in \mathcal{R}_\Sigma$ is a different credulous system, formed from a given standard default system Σ. Our goal is to determine which of these representation systems is the closest to $\mathcal{O}_{qa}[\,\cdot\,]$. To quantify this, we first define an "accuracy function" $c(\cdot,\,\cdot)$, where $c(\mathrm{R},\,q)$ quantifies the quality of the answer provided by the representation system R to the query q:

$$c(\mathrm{R},\,q) \quad \stackrel{def}{=} \quad \begin{cases} 1 & \text{if } \mathrm{R}(q) = \mathcal{O}_{qa}[\,q\,] \\ \frac{1}{2} & \text{if } \mathrm{R}(q) = \texttt{IDK} \\ 0 & \text{otherwise} \end{cases}$$

Hence, $c(\mathrm{R}_1,\,\text{“}2+2 = ?x\text{”}) = 1$ as R_1 provides the correct answer here $c(\,\mathrm{R}_1,\,\text{“P = NP”}) = 1/2$ as R_1 is silent on this question, and $c(\mathrm{R}_2,\,\text{“}2+2 = 7\text{”}) = 0$ as R_2 provides an incorrect answer.

Hence, $c(\mathrm{R},\,q)$ measure R's accuracy for a single query q. In general, we expect our representation system to deal with a range of queries. We model this using a given stationary probability function, $P : \mathcal{Q} \mapsto [0, 1]$, where $P[q]$ is the probability that the query q will occur.[5] Given this distribution, we can compute the "expected accuracy" of each system,

$$\mathrm{C}[\,\mathrm{R}\,] \;=\; E[\,c(\mathrm{R},\,\mathbf{q})\,] \;=\; \sum_{q \in \mathcal{Q}} P[q] \times c(\mathrm{R},\,q)\;. \tag{1}$$

Our challenge is to find the system R_{opt} in $\mathcal{R}$ whose expected accuracy is optimal; i.e.,

$$\text{find } \; \mathrm{R}_{opt} \in \mathcal{R}_\Sigma \; \text{ such that } \; \forall \mathrm{R} \in \mathcal{R}_\Sigma,\; \mathrm{C}[\,\mathrm{R}_{opt}\,] \;\geq\; \mathrm{C}[\,\mathrm{R}\,]\;.$$

2.2 Prioritized THEORIST-Style Representation Systems

While much of our analysis applies to representation systems in general, this paper focuses one particular form: stratified THEORIST-style representation system [PGA86] [Bre89, vA90]. Here, each R_i can be expressed as a set of factual information, a set of allowed hypotheses (each a simple type of default) and an

[5] We assume $\mathcal{Q}$ is at most countably infinite to simplify the presentation, and to avoid measure-theoretic technicalities.

ordering of the hypotheses. As a specific example, consider $\mathrm{R}_A = \langle \mathcal{F}_0, \mathcal{H}_0, \Upsilon_A \rangle$, where

$$\mathcal{F}_0 \quad = \quad \left\{ \begin{array}{l} \forall x.\ \mathtt{E}(x)\ \&\ \mathtt{N}_E(x) \Rightarrow \mathtt{S}(x,\ \mathtt{G}) \\ \forall x.\ \mathtt{A}(x)\ \&\ \mathtt{N}_A(x) \Rightarrow \mathtt{S}(x,\ \mathtt{W}) \\ \forall x.\ \neg\mathtt{S}(x,\ \mathtt{G})\ \vee\ \neg\mathtt{S}(x,\ \mathtt{W}) \\ \mathtt{A(Z)},\ \mathtt{E(Z)},\ \ldots \end{array} \right\} \tag{2}$$

is the fact set;

$$\mathcal{H}_0 \quad = \quad \left\{ \begin{array}{l} h_1 : \mathtt{N}_E(x) \\ h_2 : \mathtt{N}_A(x) \end{array} \right\}$$

is the hypothesis set, and and $\Upsilon_A = \langle h_1,\ h_2 \rangle$ is the hypothesis ordering.[6]

To explain how R_A would process a query, imagine we want to know the color of Zelda — i.e., we want to find a binding for `?c` such that $\sigma =$ "`S(Z, ?c)`" holds. R_A would first try to prove σ from the factual information $\mathcal{F}_0$ alone. This would fail, as we cannot prove that `Zelda` is a normal elephant nor that she is a normal albino (as neither $\mathtt{N}_E(\mathtt{Zelda})$ nor $\mathtt{N}_A(\mathtt{Zelda})$ hold, respectively). R_A then considers using some hypothesis — i.e., it may assert an instantiation of some element of $\mathcal{H}_0$ if that proposition is both consistent with the known facts $\mathcal{F}_0$ and also allows us to reach a conclusion to the query posed. Here, R_A could consider asserting either $\mathtt{N}_E(Z)$ (meaning that Zelda is a "normal" elephant and hence is colored *G*ray) or $\mathtt{N}_A(Z)$ (meaning that Zelda is a "normal" albino and hence is colored *W*hite). Notice that either of these options, individually, is consistent with everything we know, as encoded by $\mathcal{F}_0$. Unfortunately, we cannot assume both options, as the resulting theory, $\mathcal{F}_0 \cup \{\, \mathtt{N}_E(Z),\ \mathtt{N}_A(Z) \,\}$ would be inconsistent.

We must, therefore, decide between these options. R_A's hypothesis ordering, Υ_A, specifies the priority of the hypotheses; here $\Upsilon_A = \langle h_1, h_2 \rangle$ means that $h_1 :\ \mathtt{N}_E(x)$ takes priority over $h_2 :\ \mathtt{N}_A(x)$, which means that R_A will return the conclusion associated with $\mathtt{N}_E(Z)$ — i.e., Gray, encoded by $\mathtt{Yes}[\mathtt{?c} \mapsto G]$, as $\mathcal{F}_0 \cup \{\mathtt{N}_E(Z)\} \models \mathtt{S(Z,\ G)}$.[7]

Now consider the $\mathrm{R}_B = \langle \mathcal{F}_0, \mathcal{H}_0, \Upsilon_B \rangle$ representation system, which differs from R_A only in terms of its hypothesis ordering: As R_B's $\Upsilon_B = \langle h_2, h_1 \rangle$ considers the hypotheses in the opposite order, it will assert that Zelda is a normal albino (i.e., $\mathtt{N}_A(Z)$) and so will return the answer $\mathtt{Yes}[\mathtt{?c} \mapsto \mathtt{W}]$ to this query; i.e., it would claim that Zelda is white.

Which of these two systems is better? If we were only concerned with this single Zelda query, then the better (i.e., "more accurate") R_i is the one with the larger value for $c(\mathrm{R}_i,\ \mathtt{S(Z,\ ?c)})$ — i.e., the R_i for which $\mathrm{R}_i(\mathtt{S(Z,\ ?c)}) = \mathcal{O}_{qa}[\mathtt{S(Z,\ ?c)}]$.

In general, however, we will have to consider a less-trivial distribution of queries. To illustrate this, imagine the "…" shown in Equation 2 corresponds to

[6] Here `Z` refers to Zelda, $\mathtt{A}(\chi)$ means χ is an albino, $\mathtt{E}(\chi)$ means χ is an elephant. The first three statements of Equation 2 state that normal elephants are gray, normal albinos are white, and (in effect) that `S` is a function.

[7] This uses the instantiation $\mathtt{S(Z,\ G)} = \mathtt{S(Z,\ ?c)/Yes}[\mathtt{?c} \mapsto G]$. We will also view "$q/\mathtt{No}$" as "$\neg q$".

$\{$`A(Z`$_1$`)`, `E(Z`$_1$`)`$, \ldots,$ `A(Z`$_{100}$`)`, `E(Z`$_{100}$`)`$\}$, stating that each $\mathtt{Z}_i$ is an albino elephant; and the distribution of queries are taken from "$\mathtt{S(Z}_i\mathtt{, ?c)}$", for various $\mathtt{Z}_i$s.

Now which R_i is better? Knowing only the color of Zelda no longer answers this question; we must also know the actual colors of the other albino elephants. In general, we must know the distribution of queries P (i.e., how often each "$\mathtt{S(Z}_i\mathtt{, ?c)}$" query is posed) and moreover, know the correct answers for each (i.e., for which $\mathtt{Z}_i$s the oracle returns $\mathcal{O}_{qa}[\,\mathtt{S(Z}_i\mathtt{, ?c)}\,] = \mathtt{Yes[?c} \mapsto \mathtt{W]}$ as opposed to $\mathcal{O}_{qa}[\,\mathtt{S(Z}_i\mathtt{, ?c)}\,] = \mathtt{Yes[?c} \mapsto \mathtt{G]}$, or some other answer). From this, we can (using Equation 1) compute the expected accuracy of each system. We can then compare these two values, $\mathrm{C}[\,\mathrm{R}_A\,]$ and $\mathrm{C}[\,\mathrm{R}_B\,]$, and select the R_i system with the larger $\mathrm{C}[\,\cdot\,]$ value.

In general, a prioritized default system $\mathrm{R} = \langle \mathcal{F}, \mathcal{H}, \Upsilon \rangle$ can contain a much larger set of hypotheses $\mathcal{H}$. The ordering Υ continues to specify the order in which to consider the hypotheses. We view it as a simple ordered sequence of the elements in $\mathcal{H}$, with the understanding that R will consider each hypothesis, one at a time in this order, until finding one that is both consistent with the underlying fact set $\mathcal{F}$, and provides an answer to the given query. To be more precise, write $\Upsilon = \langle h_1, \ldots h_n \rangle$, and let i be the smallest index such that $\mathrm{Consist}(\mathcal{F} \cup \{h_i\})$ and $\mathcal{F} \cup \{h_i\} \models q/\lambda$ for some answer λ (which is either $\mathtt{Yes}[\cdots]$ or `No`); here R returns this λ. If there are no such i's, then R will return `IDK`. (Subsection 4.2 discusses how to extend this approach, to handle more general contexts.)

Our basic goal is to find the hypothesis ordering whose expected accuracy is maximal. Unfortunately, there are two major obstacles that prevent us from attaining this goal in practice:

1. The expected accuracy of any ordering depends critically on the natural distribution over queries occurring in the domain. It is unlikely that this information will be known *a priori*.
2. Even if we knew this distribution, the task of identifying the optimal hypothesis ordering is NP-complete. This holds even for the simplistic situation we have been considering, where every derivation requires exactly one hypothesis, every ordering of hypotheses is allowed, and so forth; see [Gre93].

3 The OptAcc Algorithm

This section presents a learning system, OptAcc, that side-steps the two problems mentioned above. OptAcc copes with the problem of an unknown query distribution by using a set of sample query/answer pairs to estimate the distribution; and copes with the intractability of finding the globally optimal hypothesis ordering by hill-climbing from a given initial ordering to a new one that is, with high probability, close to a local optimum. Here, by accepting a near locally optimal solution with high probability (rather than insisting on achieving a globally optimal solution with certainty), we obtain a system that can effectively produce a practical, useful result, even when the underlying domain statistics are not known *a priori*. This section first overviews OptAcc's behavior and shows its code, then states the fundamental theorem that specifies its functionality. Section 4 then presents several extensions to the algorithm.

OptAcc takes as arguments an initial representation system (read "prioritized default theory") $\mathrm{R}_0 = \langle \mathcal{F}, \mathcal{H}, \Upsilon_0 \rangle$ along with parameters $\epsilon, \delta > 0$. Each possible ordering Υ_i of the set of hypotheses $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$ corresponds to a different representation system $\mathrm{R}_i = \langle \mathcal{F}, \mathcal{H}, \Upsilon_i \rangle$. This set of alternative representation systems can be organized into a search space by specifying a set of transformation functions between orderings, thus imposing a neighborhood structure on the set. In particular, OptAcc uses a set of $O(n^2)$ possible transformations $\mathcal{T} = \{\tau_{i,j}\}_{1 \leq i,j \leq n}$, where each $\tau_{i,j}$ maps orderings to orderings: Given any ordering $\Upsilon = \langle h_1, h_2, \ldots, h_n \rangle$,

$$\tau_{i,j}(\Upsilon) = \langle h_1, \ldots, h_{i-1}, \underline{h_j}, h_i, \ldots, h_{j-1},\ h_{j+1}, \ldots, h_n \rangle$$

i.e., $\tau_{i,j}$ moves the j^{th} term in the hypothesis sequence to just before the i^{th} term. The set $\mathcal{T}[\Upsilon] = \{\, \tau_{i,j}(\Upsilon) \,\}_{i,j}$ defines the set of Υ's neighbors. Notice these transformations fully connect our space of representation systems.

Algorithm OptAcc($\langle \mathcal{F}, \mathcal{H}, \Upsilon_0 \rangle$, ϵ, δ)

  Let $K = \lceil \frac{2}{\epsilon} \rceil$

  For $k = 0 \,..\, (K-1)$ **do**

    Let $\mathcal{T}[\Upsilon_k] \leftarrow \{ \tau(\Upsilon_k) \in \mathcal{R}_{\langle \mathcal{F}, \mathcal{H} \rangle} \mid \tau \in \mathcal{T} \}$,

$$L_k \leftarrow \left\{ \begin{array}{ll} \left\lceil \frac{8}{\epsilon^2} \ln \frac{2\,K\,(1+|\mathcal{T}[\Upsilon_0]|)}{\delta} \right\rceil & \text{if } k = 0 \\ \left\lceil \frac{8}{\epsilon^2} \ln \frac{2\,K\,|\mathcal{T}[\Upsilon_k]|}{\delta} \right\rceil & \text{otherwise} \end{array} \right\}$$

    Draw L_k sample queries from the $P[\cdot]$ distribution, $S_k = \{q_1, \ldots, q_{L_k}\}$

    ForEach $\Upsilon' \in \mathcal{T}[\Upsilon_k]$ **do**

      Let $\hat{\mathrm{C}}[\,\Upsilon'\,] \leftarrow \frac{1}{L} \sum_{i=1}^{L} c(\Upsilon', q)$.

    (**If** $k = 0$, **then**

      Let $\hat{\mathrm{C}}[\,\Upsilon_0\,] \leftarrow \frac{1}{L} \sum_{i=1}^{L} c(\Upsilon_0, q)$.)

    If $\exists \Upsilon' \in \mathcal{T}[\Upsilon_k]$ s.t. $\hat{\mathrm{C}}[\,\Upsilon'\,] > \hat{\mathrm{C}}[\,\Upsilon_k\,] + \frac{\epsilon}{2}$

      Then **Let** $\Upsilon_{k+1} \leftarrow \Upsilon'$

      Else **Return**[Υ_k].

  End For

End OptAcc

Fig. 1. Code for OptAcc

OptAcc's code appears in Figure 1. In essence, OptAcc will climb from Υ_k to one of its neighbors, $\Upsilon' \in \mathcal{T}[\Upsilon_k]$, if this Υ' is statistically likely to be superior to Υ_k; i.e., if we are highly confident that $\mathrm{C}[\Upsilon_{k+1}] > \mathrm{C}[\Upsilon_k]$.[8] This constitutes one hill-climbing step; in general, OptAcc will perform many such steps, climbing from Υ_0 to Υ_1 to Υ_2, and so on, until terminating on reaching Υ_m, for some $m \leq K$. Here, we are confident that none of Υ_m's neighbors $\mathcal{T}[\Upsilon_m]$ is more than ϵ better than Υ_m. Theorem 1 specifies OptAcc's behavior more precisely; its proof appears in the appendix.

Theorem 1. *The* OptAcc$(\langle \mathcal{F}, \mathcal{H}, \Upsilon_0 \rangle, \epsilon, \delta)$ *algorithm incrementally produces a series of hypothesis orderings* $\Upsilon_0, \Upsilon_1, \ldots, \Upsilon_m$ *such that, with probability at least* $1-\delta$*, both*

1. *the expected accuracy of each successive ordering in the series is strictly better than its predecessors'; i.e.,*

$$\forall i > j,\ \mathrm{C}[\Upsilon_i] \ > \ \mathrm{C}[\Upsilon_j]$$

2. *the final ordering* Υ_m *in the series is an "*ϵ*-local optimum"; i.e.,*

$$\forall \tau \in \mathcal{T},\ \mathrm{C}[\Upsilon_m] \ \geq \ \mathrm{C}[\ \tau(\Upsilon_m)\ ] - \epsilon\ .$$

Moreover, OptAcc *requires only a number of query/answer samples that is polynomial in* $1/\epsilon$, $1/\delta$ *and* $|\mathcal{H}|$. □

4 Issues and Extensions

This section discusses: other algorithms related to OptAcc, ways for OptAcc to accommodate more general Theorist-style representations, efficiency issues, and alternative performance measures and types of transformations.

4.1 Related Algorithms

We can view OptAcc as a variant on *anytime algorithms* [BD88, DB88] as, at any time, OptAcc provides a usable result (here, the theory produced at the k^{th} iteration, Υ_k), with the property that later systems are (probably) better than earlier ones; i.e., $i > j$ means $\mathrm{C}[\Upsilon_i] > \mathrm{C}[\Upsilon_j]$ with high probability. OptAcc differs from standard anytime algorithms by terminating on reaching a point of diminishing returns.

OptAcc works in a "batched incremental" mode, as it iteratively uses a *set* of samples to decide whether to climb to a new theory, or to terminate. There is also a strictly-incremental variant of this algorithm [Gre92b], which observes samples one-by-one, and decides after each individual sample, whether to climb, terminate, or simply draw an additional sample; hence this variant can, in some situations, climb to better theories after fewer samples.

[8] Here, as in Figure 1, "$\mathrm{C}[\Upsilon_\alpha]$" refers to "$\mathrm{C}[\langle \mathcal{F}, \mathcal{H}, \Upsilon_\alpha \rangle]$"; "$c(\Upsilon_\alpha, q)$" refers to "$c(\langle \mathcal{F}, \mathcal{H}, \Upsilon_\alpha \rangle, q)$"; and $\mathcal{R}_{\langle \mathcal{F}, \mathcal{H} \rangle}$ refers to the set of all credulous default theories formed from the underlying standard default theory $\langle \mathcal{F}, \mathcal{H} \rangle$.

4.2 Accommodating More General THEORIST-Style Representations

The descriptions above have assumed that every ordering of hypotheses is meaningful. In some contexts, there may already be a meaningful partial ordering of the hypotheses, perhaps based on specificity or some other criteria [Gro91]. Here, we can still use OPTACC to complete the partial ordering, by determining the relative priorities of the initially incomparable elements.

In some situations, we may be unable to answer certain queries without adding in *several* new assertions. We can model this by viewing $\mathcal{H} = \mathcal{P}[H]$ as the power set of some set of "sub-hypotheses", H. If we then define orderings on the hypotheses $\mathcal{H}$ that correspond to lexographic extensions of orderings over H, we can then move about this subset of $\mathcal{H}$-orderings by simply modifying H-orderings.

4.3 Efficiency

As OPTACC must determine whether $\mathcal{F} \cup \{h_i\} \models^? q/\mathcal{O}_{qa}[q]$, it can require general theorem proving. This derivation process is the critical factor in determining OPTACC's computation cost: if the derivation process is decidable (e.g., if we are dealing with propositional theories), then OPTACC will necessarily terminate; and if it is polytime (e.g., if we are dealing with propositional Horn theories or propositional 2-CNF), that the OPTACC algorithm will be polytime.

Notice next that OPTACC requires the values of $\hat{C}[\Upsilon'] = \sum_{q \in S} c(\Upsilon', q)$ for each $\Upsilon' \in \mathcal{T}[\Upsilon_k]$. We can, in general, obtain this information by determining whether $\mathcal{F} \cup \{h_i\} \models^? q/\mathcal{O}_{qa}[q]$ holds for each hypothesis h_i. There can, in some situations, be more efficient ways of estimating these values, for example, by using some Horn approximation to $\mathcal{F} \cup \{h_i\}$; see [Gre92a] and [GJ92]. We can also simplify the computation if the h_j hypotheses are not independent; e.g., if each corresponds to a set of sub-hypotheses.

4.4 Alternative Performance Measures and Transformations

We have so far insisted that each categorical answer to a query be either completely correct or completely false; in general, we can imagine a range of answers to a query, some of which are better than others. (Imagine for example that the correct answer to a particular existential query is a set of 10 distinct instantiations. Here, returning 9 of them may be better than returning 0, or than returning 1 wrong answer. As another situation, we may be able to rank responses in terms of their precision: e.g., knowing that the cost of `watch`$_7$ is `$3,000` is more precise than knowing only that `watch`$_7$ is `expensive` [Vor91].) We have also assumed that all queries are equally important; i.e., a wrong answer to any query "costs" us the same 0, whether we are asking for the location of a salt-shaker, or of the tiger currently stalking us.

One way of addressing all of these points is to use a more general $c(\mathrm{R}, q)$ function — one that can incorporate these different factors, by differentially weighting the different queries, the different possible answers, etc. In fact, we could permit the user to specify his own $c(\mathrm{R}, q)$ function.

Notice also that we have completely discounted the computational cost associated with arriving at the answer. Within this framework, we can consider yet more general $c(\cdot, \cdot)$ "utility functions", which can even incorporate the user's tradeoffs among accuracy, categoricity, efficiency, and perhaps other aspects. This would allow the user to prefer, for example, a performance system that returns `IDK` in complex situations, rather than spend a long time returning the correct answer; or even allow it to be wrong in some instances [GE91].

Of course, the OptAcc-variant may have to consider other transformations, besides the simple "reordering the hypotheses" one discussed above. For example, if being wrong was much worse than being silent (i.e., returning "`IDK`"), we could transform one representation system to another by including a rule whose conclusion is `IDK`, which applies in certain cases where the correct answer is not known reliably. Such a system might, perhaps, include $h_3 :\ \mathtt{N}_{AE}(x)$ in its hypothesis set and include the rule $\forall x.\ \mathtt{A}(x)\ \&\ \mathtt{E}(x)\ \&\ \mathtt{N}_{AE}(x) \Rightarrow \mathtt{S}(x, \mathtt{IDK})$ in its fact set. A representation system that accepts this $\mathtt{N}_{AE}(\mathtt{Z}_{15})$ hypothesis will produce the answer `IDK` to the query $\mathtt{S}(Z_{15}, ?y)$.

There are yet other types of transformations, for converting one representation system into another — for instance eliminating some inappropriate sets of hypotheses [Coh90, Won91], or modifying the antecedents of individual rules (*cf.*, [OM90]), etc. Each of these approaches can be viewed as using a set of transformations to navigate around a space of interrelated representation systems. We can then consider the same objective described above: to identify which element has the highest expected accuracy (or in general, "highest expected utility").

Here, as above, the expected utility score for each element depends on the unknown distribution, meaning we will need to use some sampling process. In some simple cases, we may be able to identify (an approximation to) the *globally* optimal element with high probability (*à la* the PAO algorithm discussed in [OG90, GO91]). In most cases, however, this identification task is intractable. Here again it makes sense to use a hill-climbing system (similar to OptAcc) to identify an element that is close to a local optimum, with high probability. (Of course, this local optimality will be based on the classes of transformations used to define the space of representation systems.)

5 Conclusion

Many specifications of nonmonotonic theories are ambiguous, in that they sanction many individually plausible but collectively incompatible solutions to certain queries; this is the essence of the multiple extension problem. This report addresses this problem by considering the set of credulous reasoning systems derived from a given nonmonotonic theory (each formed by imposing a total ordering on the hypotheses) and then attempting to identify the credulous system that is correct most often — i.e., which has the highest "expected accuracy", with respect to the distribution of queries and correct answers. Unfortunately, the natural distribution of queries is usually not known *a priori*, and moreover, the task of identifying the optimal system is intractable, even given this distribution. We present a learning algorithm, OptAcc, that side-steps these problems by using a set of query/answer samples to obtain an estimate of the unknown distribution, and by using a set of transformations to hill-climb to a credulous system that is, with high probability, arbitrarily close to a local optimum. We also show that this algorithm is efficient, in that its sample complexity is only a low-order polynomial in the size of the initial theory and the (reciprocal) error and confidence terms; and its computational complexity is dominated by the cost of the underlying derivation process.

Acknowledgements

Some of this work was performed at the University of Toronto, where the first author was supported by the Institute for Robotics and Intelligent Systems, and by an operating grant from the National Science and Engineering Research Council of Canada. Both authors gratefully acknowledge receiving many helpful comments from William Cohen, Charles Elkan and Jonathan Wong.

A Proof of Theorem 1

Theorem 1 *The* OptAcc$(\langle \mathcal{F}, \mathcal{H}, \Upsilon_0 \rangle,\ \epsilon,\ \delta)$ *algorithm incrementally produces a series of orderings* $\Upsilon_0, \Upsilon_1, \ldots, \Upsilon_m$ *such that, with probability at least* $1-\delta$*, both*

1. *the expected accuracy of each successive ordering in the series is strictly better than its predecessor's; i.e.,*
$$\forall i > j,\ \ C[\,\Upsilon_i\,] \ > \ C[\,\Upsilon_j\,]$$
2. *the final ordering* Υ_m *in the series is an "ϵ-local optimum"; i.e.,*
$$\forall \tau \in \mathcal{T},\ \ C[\,\Upsilon_m\,] \ \geq \ C[\ \tau(\Upsilon_m)\ ] - \epsilon\ .$$

Moreover, OptAcc *requires only a number of query/answer samples that is polynomial in* $1/\epsilon$*,* $1/\delta$ *and* $|\mathcal{H}|$*.*

Proof: To deal with OptAcc's efficiency: Notice that it will stay at any Υ_k performance element for L_k samples, a quantity that is clearly polynomial in $|\mathcal{T}[\Upsilon_j]| = O(|\mathcal{H}|^2)$, $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$. Also observe that OptAcc can climb at most $K-1$ times: It will only climb from Υ_k to a new Υ_{k+1} if the empirical estimate $\hat{\mathrm{C}}[\,\Upsilon_{k+1}\,]$ is at least $\epsilon/2$ over $\hat{\mathrm{C}}[\,\Upsilon_k\,]$; hence, after ℓ climbs, $\hat{\mathrm{C}}[\,\Upsilon_\ell\,] \geq \hat{\mathrm{C}}[\,\Upsilon_0\,] + \ell\epsilon/2$. After $K-1$ climbs, the empirical average of the resulting Υ_{K-1} is at least $\hat{\mathrm{C}}[\,\Upsilon_{K-1}\,] \ \geq\ \hat{\mathrm{C}}[\,\Upsilon_0\,] \ +\ (K-1)\frac{\epsilon}{2} \ \geq\ 0 \ +\ (\frac{2}{\epsilon}-1)\frac{\epsilon}{2} \ =\ 1-\frac{\epsilon}{2}$. As $\hat{\mathrm{C}}[\,\Upsilon\,]$ can be at most 1 for any theory, no theory can be strictly more than $\frac{\epsilon}{2}$ better than this Υ_{K-1} theory, and so there can be no additional climbs.

To prove Parts 1 and 2, notice there are two types of mistakes that OptAcc can make on a single stage of the OptAcc algorithm, when it is dealing with Υ_k:

A_k. OptAcc climbed from Υ_k to some $\Upsilon' = \tau(\Upsilon_k)$ as Υ' appeared to be better than Υ_k, but in reality, Υ' was not better; or

B_k. OptAcc terminated as no $\Upsilon' = \tau(\Upsilon_k)$ appeared to be more than ϵ better than Υ_k, but there was some such Υ' that is much better.

Notice that neither A_k nor B_k can occur if $\hat{C}[\Upsilon'] = \frac{1}{|S_k|}\sum_{q \in S_k} c(\Upsilon', q)$, the empirical estimate of $C[\Upsilon']$ obtained using the samples S_k, is within $\frac{\epsilon}{4}$ of the $C[\Upsilon']$ for each relevant Υ'; i.e., if

$$\forall \Upsilon' \in \{\Upsilon_k\} \cup \mathcal{T}[\Upsilon_k], \quad \left|\hat{C}[\Upsilon'] - C[\Upsilon']\right| \;\leq\; \frac{\epsilon}{4}\,. \tag{3}$$

(Proof: For A_k, if $\hat{C}[\Upsilon'] > \hat{C}[\Upsilon_k] + \frac{\epsilon}{2}$, then

$$\begin{aligned} C[\Upsilon_k] - C[\Upsilon'] &= (C[\Upsilon_k] - \hat{C}[\Upsilon_k]) + \\ &\qquad (\hat{C}[\Upsilon_k] - \hat{C}[\Upsilon']) \;+\; (\hat{C}[\Upsilon'] - C[\Upsilon']) \\ &< \tfrac{\epsilon}{4} + -\tfrac{\epsilon}{2} + \tfrac{\epsilon}{4} \;=\; 0 \end{aligned}$$

and for B_k, if $\hat{C}[\Upsilon'] \leq \hat{C}[\Upsilon_k] + \frac{\epsilon}{2}$, then

$$\begin{aligned} C[\Upsilon'] - C[\Upsilon_k] &= (C[\Upsilon'] - \hat{C}[\Upsilon']) + \\ &\qquad (\hat{C}[\Upsilon'] - \hat{C}[\Upsilon_k]) \;+\; (\hat{C}[\Upsilon_k] - C[\Upsilon_k]) \\ &\leq \tfrac{\epsilon}{4} + \tfrac{\epsilon}{2} + \tfrac{\epsilon}{4} \;=\; \epsilon\,. \end{aligned}$$)

We therefore need only show that Equation 3 holds with probability at least $1 - \frac{\delta}{K}$, as that means that the probability of making either type of mistake on the k^{th} iteration is at most $\frac{\delta}{K}$, and so the total probability that OptAcc will make any mistake, on any iteration, is at most $K\frac{\delta}{K} = \delta$.

These claims follow immediately from Hoeffding's inequality (a variant of Chernoff bounds): As each query q_i is selected independently from a fixed distribution, the values $\{c(\Upsilon, q_i)\}_i$ are independent, identically-distributed random values. Hoeffding's inequality states that their observed sample average, $\frac{1}{|S|}\sum_{q_i \in S} c(\Upsilon, q_i) = \hat{C}[\Upsilon']$, converges exponentially fast to the population mean, $C[\Upsilon]$: i.e., the probability that "$\hat{C}[\Upsilon]$ is not within γ of $C[\Upsilon]$" goes to 0 exponentially fast as $|S|$ increases; and, for a fixed $|S|$, exponentially as γ increases. Formally,[9]

$$Pr\left[\left|\hat{C}[\Upsilon'] - C[\Upsilon']\right| \geq \gamma\right] \quad\leq\quad 2e^{-2|S|\gamma^2}\,. \tag{4}$$

In the $k = 0$ situation, as OptAcc uses $L_0 = \left\lceil \frac{8}{\epsilon^2} \ln \frac{2K(1+|\mathcal{T}[\Upsilon_0]|)}{\delta} \right\rceil$ samples S_0, the probability that any $\hat{C}[\Upsilon']$ is *not* within $\epsilon/4$ of $C[\Upsilon']$ (for any theory $\Upsilon' = \Upsilon_0$ or $\Upsilon' \in \mathcal{T}[\Upsilon_0]$) is

$$\begin{aligned} Pr\left[\left|\hat{C}[\Upsilon'] - C[\Upsilon']\right| > \tfrac{\epsilon}{4}\right] &\leq 2e^{-2L_0\left(\frac{\epsilon}{4}\right)^2} \\ &\leq 2e^{-2\frac{8}{\epsilon^2}\ln\frac{2K(1+|\mathcal{T}[\Upsilon_0]|)}{\delta}\frac{\epsilon^2}{16}} \\ &= 2\frac{\delta}{2K(1+|\mathcal{T}[\Upsilon_0]|)} \quad=\quad \frac{\delta}{K(1+|\mathcal{T}[\Upsilon_0]|)}\,. \end{aligned}$$

[9] See [Bol85, p. 12]. *N.b.*, these inequalities hold for arbitrary *bounded* random variables, and thus for $\hat{C}[\Upsilon']$ as $0 \leq c(\Upsilon', q_i) \leq 1 \;\forall q_i \in \mathcal{Q}$.

Hence, the probability that any of the $1+|\mathcal{T}[\Upsilon_0]|$ estimates $\hat{\mathrm{C}}[\Upsilon']$ is not within $\epsilon/4$ of the corresponding $\mathrm{C}[\Upsilon']$ is $(1+|\mathcal{T}[\Upsilon_0]|)\frac{\delta}{K\,(1+|\mathcal{T}[\Upsilon_0]|)} = \frac{\delta}{K}$.

Now consider any $k \geq 1$, and observe that we have already obtained the estimate $\hat{\mathrm{C}}[\Upsilon_k]$ during the $k-1^{st}$ stage, and are already confident that it is within $\epsilon/4$ of $\mathrm{C}[\Upsilon_k]$. We therefore need only show that our estimates of the accuracy of each $\Upsilon' \in \mathcal{T}[\Upsilon_k]$ is within $\epsilon/4$ of the correct value; this again follows trivially from the Equation 4 and the fact that OptAcc draws $L_k = \left\lceil \frac{8}{\epsilon^2} \ln \frac{2\,K\,|\mathcal{T}[\Upsilon_k]|}{\delta} \right\rceil$ samples. □ (Theorem 1)

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.

[BD88] Mark Boddy and Thomas Dean. Solving time dependent planning problems. Technical report, Brown University, 1988.

[BE89] Alex Borgida and David Etherington. Hierarchical knowledge bases and efficient disjunctive reasoning. In *Proceedings of KR-89*, pages 33–43, Toronto, May 1989.

[BMSJ78] Bruce G. Buchanan, Thomas M. Mitchell, Reid G. Smith, and C. R. Johnson, Jr. Models of learning systems. In *Encyclopedia of Computer Science and Technology*, volume 11. Dekker, 1978.

[Bol85] B. Bollobás. *Random Graphs*. Academic Press, 1985.

[Bre89] Gerhard Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *Proceedings of IJCAI-89*, pages 1043–48, Detroit, August 1989.

[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.

[DB88] Thomas Dean and Mark Boddy. An analysis of time-dependent planning. In *Proceedings of AAAI-88*, pages 49–54, August 1988.

[DE92] Mukesh Dalal and David Etherington. Tractable approximate deduction using limited vocabulary. In *Proceedings of CSCSI-92*, Vancouver, May 1992.

[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.

[FD89] N. Flann and T. G. Dietterich. A study of explanation-based methods for inductive learning. *Machine Learning*, 4, 1989.

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

[GE91] Russell Greiner and Charles Elkan. Measuring and improving the effectiveness of representations. In *Proceedings of IJCAI-91*, pages 518–24, Sydney, Australia, August 1991.

[GJ92] Russell Greiner and Igor Jurišica. A statistical approach to solving the EBL utility problem. In *Proceedings of AAAI-92*, San Jose, 1992.

[GO91] Russell Greiner and Pekka Orponen. Probably approximately optimal derivation strategies. In J.A. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of KR-91*, San Mateo, CA, April 1991. Morgan Kaufmann.

[Gre92a] Russell Greiner. Learning near optimal horn approximations. In *Proceedings of Knowledge Assimilation Symposium*, Stanford, March 1992.

[Gre92b] Russell Greiner. Probabilistic hill-climbing: Theory and applications. In *Proceedings of CSCSI-92*, Vancouver, June 1992.

[Gre93] Russell Greiner. The complexity of computing optimally-accurate default theories. Technical report, Siemens Corporate Research, 1993.

[Gro91] Benjamin Grosof. Generalizing prioritization. In *Proceedings of KR-91*, pages 289–300, Boston, April 1991.

[GS92] Russell Greiner and Dale Schuurmans. Learning useful horn approximations. In B. Nebel, C. Rich, and W. Swartout, editors, *Proceedings of KR-92*, San Mateo, CA, October 1992. Morgan Kaufmann.

[Hau88] David Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, pages 177–221, 1988.

[Hin89] Geoff Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1-3):185–234, September 1989.

[HV88] David Haussler and Leslie Valiant, editors. *Proceedings of the First Workshop on Computational Learning Theory.* Morgan Kaufmann, MIT, 1988.

[Kyb82] H. Kyburg. The reference class. *Philosophy of Science*, 50, 1982.

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

[Lou88] R. Loui. Computing reference classes. In *AAAI Workshop on Uncertainty.* Morgan Kaufmann, St Paul, 1988.

[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.

[MCM83] Ryszard S. Michalski, Jaime G. Carbonell, and Thomas M. Mitchell, editors. *Machine Learning: An Artificial Intelligence Approach.* Tioga Publishing Company, Palo Alto, CA, 1983.

[Mit80] Thomas M. Mitchell. The need for bias in learning generalizations. Technical Report CBM-TR-117, Laboratory for Computer Science Research, May 1980.

[Mor87] Paul Morris. Curing anomalous extensions. In *Proceedings of AAAI-87*, pages 437–42, Seattle, July 1987.

[OG90] Pekka Orponen and Russell Greiner. On the sample complexity of finding good search strategies. In *Proceedings of COLT-90*, pages 352–58, Rochester, August 1990.

[OM90] Dirk Ourston and Raymond J. Mooney. Changing the rules: A comprehensive approach to theory refinement. In *Proceedings of AAAI-90*, pages 815–20, 1990.

[Paz88] M. Pazzani. Selecting the best explanation in explanation-based learning. In *Proceedings of Symposium on Explanation-Based Learning*, Stanford, March 1988.

[PGA86] David Poole, Randy Goebel, and Romas Aleliunas. Theorist: A logical reasoning system for default and diagnosis. Technical Report CS-86-06, Logic

Programming and Artificial Intelligence Group, Faculty of Mathematics, University of Waterloo, February 1986.

[Qui90] J. Ross Quinlan. Learning logical definitions from relations. *Machine Learning Journal*, 5(3):239–66, August 1990.

[Rei87] Raymond Reiter. Nonmonotonic reasoning. In *Annual Review of Computing Sciences*, volume 2, pages 147–87. Annual Reviews Incorporated, Palo Alto, 1987.

[RG87] Stuart J. Russell and Benjamin N. Grosof. A declarative approach to bias in concept learning. In *Proceedings of AAAI-87*, pages 505–10, Seattle, WA, July 1987.

[Sha83] Ehud Shapiro. *Algorithmic Program Debugging*. MIT Press, 1983.

[Sha89] Lokendra Shastri. Default reasoning in semantic networks: A formalization of recognition and inheritance. *Artificial Intelligence*, 39:283–355, 1989.

[SK91] Bart Selman and Henry Kautz. Knowledge compilation using horn approximations. In *Proceedings of AAAI-91*, pages 904–09, Anaheim, August 1991.

[vA90] Paul van Arragon. Nested default reasoning with priority levels. In *Proceedings of CSCSI-90*, pages 77–83, Ottawa, May 1990.

[Vor91] David Vormittag. Evaluating answers to questions, May 1991. Bachelors Thesis, University of Toronto.

[Won91] Jonathan Wong. Improving the accuracy of a representational system, May 1991. Bachelors Thesis, University of Toronto.