Appeared in
ECAI Workshop on Theoretical Foundations of Knowledge Representation and Reasoning,
(ed: Gerhard Lakemeyer & Bernhard Nebel), Vienna, August 1992.

Learning an Optimally Accurate Representational System

Russell Greiner*
Siemens Corporate Research
755 College Road East
Princeton, NJ 08540-6632
greiner@learning.siemens.com

Dale Schuurmans
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 1A4
dale@cs.toronto.edu

Abstract

The multiple extension problem arises because a default theory can use different subsets of its defaults to propose different, mutually incompatible, answers to some queries. This paper presents an algorithm that uses a set of observations to learn a credulous version of this default theory that is (essentially) "optimally accurate".

In more detail, we can associate a given default theory with a set of related credulous theories $\mathcal{R} = \{\mathrm{R}_i\}$, where each R_i uses its own total ordering of the defaults to determine which single answer to return for each query. Our goal is to select the credulous theory that has the highest "expected accuracy", where each R_i's expected accuracy is the probability that the answer it produces to a query will correspond correctly to the world.

Unfortunately, a theory's expected accuracy depends on the distribution of queries, which is usually not known. Moreover, the task of identifying the optimal $\mathrm{R}_{opt} \in \mathcal{R}$, even given that distribution information, is intractable. This paper presents a method, OptAcc, that sidesteps 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 parameters $\epsilon, \delta > 0$, OptAcc produces an $\mathrm{R}_{oa} \in \mathcal{R}$ whose expected accuracy is, with probability at least $1 - \delta$, within ϵ of a local optimum.

1 Introduction

A "representational 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 produces the answer "4" to the query "find x such that $2+2=x$", produces the accepted bid for each hand in bridge, finds the correct diagnosis from a given set of patient symptoms, etc. We define R's "expected accuracy" as the average accuracy of the answers it produces, over the distribution of queries posed. Our goal is to find the representational system with the highest possible expected accuracy.

Many representational 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, etc.) that is plausible but not necessarily true; each particular collection of facts and hypotheses (a.k.a. defaults) is a default theory. 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]. This is called the "multiple extension problem" in the knowledge representation community, and the "bias problem" in machine learning. In each, it has produced a great deal of attention and debate; *cf.*, [Rei87] [Mit80, RG87, Hau88].

To be useful, our representational system must return but a single answer. We therefore focus on a credulous form of such theories, formed by embellishing each default theory with an ordering on the defaults [vA90, Bre89], with the understanding that only the most preferred defaults(s) will be used to reach a unique answer to each query; see Section 2.[1]

The obvious question then arises: what is the best ordering of the defaults? We provide the obvious pragmatic answer: use the ordering that is most likely to be "correct" — *i.e.*, is "optimally accurate". Section 2 defines this correctness 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 in-

*Much of this work was performed at the University of Toronto, where it 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. We also gratefully acknowledge receiving many helpful comments from W. Cohen, C. Elkan and J. Wong.

[1] We can allow a system to remain skeptical by using highly-preferred defaults that conclude "IDK" (for "I Don't Know") in some situations; see Note6 in Section 3.

formation is usually not known *a priori*. Moreover, the task of identifying the optimal ordering, even given that distribution information, is generally intractable. Section 3 then presents a method that side-steps these problems by using a set of query/answer pairs to estimate the unknown distribution; and by hill-climbing to a local optimum. In particular, it describes the OptAcc algorithm that, given the parameters $\epsilon, \delta > 0$, returns a ordering whose expected accuracy is, with probability greater than $1 - \delta$, within ϵ of a local optimum. That section concludes by discussing some extensions to this algorithm.

2 Framework

We assume that there is an underlying set of queries that can be posed to the representational system R, and that these queries are drawn at random from this set according to some unknown but stationary distribution P. We model this using an oracle $\mathcal{O}$ that, on each call, returns a pair $\langle q, a\rangle$ where q is a query drawn at random from P and a is the "correct" answer to this query.[2] For now, we assume each correct answer is either "$\texttt{No}$" or "$\texttt{Yes}[x_i \rightsquigarrow \texttt{V}_i]$", where the mapping within the brackets is a binding list of q's free variables.[3] As examples, one call to the oracle may return the pair $\langle$'$\texttt{2+2=?x}$', $\texttt{Yes}[\texttt{?x} \rightsquigarrow \texttt{4}]\rangle$, and another, $\langle$'$\texttt{2+2=19}$', $\texttt{No}\rangle$. To simplify our presentation, we write $\mathcal{O}'[\,q\,] = a$ whenever $\mathcal{O}$ returns $\langle q,\ a\rangle$; hence $\mathcal{O}'[\,2+2 = x\,] = \texttt{Yes}[?x \rightsquigarrow \texttt{4}]$.

We say an R's answer to a query q,[4] written $\mathrm{R}(q)$, is "correct" if $\mathrm{R}(q)$ matches $\mathcal{O}'[\,q\,]$, is "incomplete" if $\mathrm{R}(q)$ is "$\texttt{IDK}$", and is "incorrect" otherwise. We will use an "accuracy function" $c(\mathrm{R}_i,\ q)$ that assigns to each such query a score of $+1$, 0, or -1, respectively:

$$c(\mathrm{R}_i,\ q) \ \overset{def}{=}\ \left\{ \begin{array}{rl} +1 & \text{if } \mathrm{R}_i(q) = \mathcal{O}'[\,q\,] \\ 0 & \text{if } \mathrm{R}_i(q) = \texttt{IDK} \\ -1 & \text{otherwise} \end{array} \right.$$

While many parts of this analysis apply in general, this paper will focus on a particular type of stratified Theorist-style representational 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 a specific ordering of the hypotheses. As a specific example, consider $\mathrm{R}_A = \langle \mathcal{F}_0,\ \mathcal{H}_0,\ \Upsilon_A\rangle$, where

$$\mathcal{F}_0 \quad = \quad \left\{ \begin{array}{l} \forall x.\ \texttt{E}(x)\ \&\ \texttt{N}_E(x) \Rightarrow \texttt{S}(x,\ \texttt{G}) \\ \forall x.\ \texttt{A}(x)\ \&\ \texttt{N}_A(x) \Rightarrow \texttt{S}(x,\ \texttt{W}) \\ \forall x.\ \neg\texttt{S}(x,\ \texttt{G})\ \vee\ \neg\texttt{S}(x,\ \texttt{W}) \\ \texttt{A(Z)},\ \texttt{E(Z)},\ \ldots \end{array} \right\} \qquad (1)$$

is the fact set;

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

is the hypothesis set, and and $\Upsilon_A = \langle h_1,\ h_2\rangle$ is the hypothesis ordering.[5]

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 $\texttt{?c}$ such that $\sigma =$ "$\texttt{S(Z, ?c)}$" holds. R_A would first try to prove σ from the factual information $\mathcal{F}_0$ alone. This would fail, as we do not know if $\texttt{Zelda}$ is a normal elephant or if she is a normal albino (*i.e.*, if $\texttt{N}_E(\texttt{Zelda})$ or $\texttt{N}_A(\texttt{Zelda})$ holds, 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 $\texttt{N}_E(Z)$ (meaning that Zelda is a "normal" elephant and hence is colored *G*ray) or $\texttt{N}_A(Z)$ (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, $\mathcal{F}_0 \cup \{\, \texttt{N}_E(Z),\ \texttt{N}_A(Z)\,\}$ is inconsistent.

We must, therefore, decide amongst 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: $\texttt{N}_E(x)$ takes priority over h_2: $\texttt{N}_A(x)$, which means that R_A will return the conclusion associated with $\texttt{N}_E(Z)$ — *i.e.*, Gray, encoded by $\texttt{Yes}[\texttt{?c} \mapsto G]$, as $\mathcal{F}_0 \cup \{\texttt{N}_E(Z)\} \models \texttt{S(Z, G)}$.[6]

Now consider the $\mathrm{R}_B = \langle \mathcal{F}_0, \mathcal{H}_0, \Upsilon_B\rangle$ representational 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 return the answer $\texttt{Yes}[\texttt{?c} \mapsto \texttt{W}]$ 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 $c(\mathrm{R}_i,\ \texttt{S(Z, ?c)})$ — *i.e.*, the R_i for which $\mathrm{R}_i(\texttt{S(Z, ?c)}) = \mathcal{O}'[\texttt{S(Z, ?c)}]$.

In general, however, we will have to consider a less-trivial distribution of queries. To illustrate this, imagine the "…" shown in Equation 1 corresponds to $\{\texttt{A}(\texttt{Z}_1),\ \texttt{E}(\texttt{Z}_1), \ldots, \texttt{A}(\texttt{Z}_{100}),\ \texttt{E}(\texttt{Z}_{100})\}$, stating that each $\texttt{Z}_i$ is an albino elephant; and the distribution of queries are taken from "$\texttt{S}(\texttt{Z}_i,\ \texttt{?c})$", for various $\texttt{Z}_i$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 "$\texttt{S}(\texttt{Z}_i,\ \texttt{?c})$" query is posed) and moreover, know the correct answers (*i.e.*, for which $\texttt{Z}_i$s the oracle returns $\mathcal{O}'[\texttt{S}(\texttt{Z}_i,\ \texttt{?c})] = \texttt{Yes}[\texttt{?c} \mapsto \texttt{W}]$ as opposed to $\mathcal{O}'[\texttt{S}(\texttt{Z}_i,\ \texttt{?c})] = \texttt{Yes}[\texttt{?c} \mapsto \texttt{G}]$, or some other answer).

[2] This oracle can be the "real world", which provides feedback to the representational system R, indicating the correctness of R's responses.

[3] Note6 in Section 3 considers a more general framework.

[4] *N.b.*, we assume that R will return a single answer. If there are several *compatible* binding lists, then R will select and return one of them; see extended paper [GS92].

[5] Here $\texttt{Z}$ refers to Zelda, $\texttt{A}(\chi)$ means χ is an albino, $\texttt{E}(\chi)$ means χ is an elephant. The first three statements of Equation 1 state that normal elephants are gray, normal albinos are white, and (in effect) that $\texttt{S}$ is a function.

[6] This uses the instantiation $\texttt{S(Z, G)} = \texttt{S(Z, ?c)}/\texttt{Yes}[\texttt{?c} \mapsto G]$. We will also view "$q/\texttt{No}$" as "$\neg q$". Note6 in Subsection 3 discusses how to produce the "$\texttt{IDK}$" answer to a query.

From this, we can compute the expected accuracy of each system,

$$\mathrm{C}[\,\mathrm{R}_i\,] \;=\; E[\,c(\mathrm{R}_i,\, \mathbf{q})\,] \;=\; \underset{P,\; q}{\mathit{average}} \;\; c(\mathrm{R}_i,\, q) \qquad (2)$$

(If there are only a finite number of queries, Equation 2 corresponds to $\mathrm{C}[\,\mathrm{R}_i\,] = \sum_q P[q] \times c(\mathrm{R}_i,\, q)$.) 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.

Everything here can scale up, to deal with more complex representational systems; in particular, R can include a much larger set of hypotheses; see Note1. The ordering Υ in $\mathrm{R} = \langle \mathcal{F}, \mathcal{H}, \Upsilon \rangle$ 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 that provides an answer to the given query. More formally, write $\Upsilon = \langle h_{\pi(1)}, \ldots h_{\pi(n)} \rangle$, and let $i = \pi(j)$ 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 `Yes`$[\cdots]$ or `No`); here R returns this λ. If there are no such i's, then R will return `IDK`. (Note1 and Note6 in Section 3 discuss how to extend this approach, to handle yet 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 query/answer pairs occurring in the domain. It is unlikely that this information would be known *a priori*.
2. Even if we knew the precise nature of 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 any derivation can involve exactly one hypothesis, every ordering of hypotheses is allowed, etc.[7]).

3 The OptAcc Algorithm

This section presents a learning system, OptAcc, that side-steps the two problems mentioned above. OptAcc accomplishes this by using a set of sample queries to estimate the distribution, and by hill-climbing from a given initial R_0 to one that is, with high probability, close to a local optimum. That is, by sacrificing our desire to achieve a globally optimal solution with certainty, and accepting a near locally optimal solution with high probability, we obtain a system that can *efficiently* produce a practical, useful result, even when the underlying domain statistics are unknown to us *a priori*. This section first states the fundamental theorem that specifies OptAcc's functionality. It then discusses OptAcc's code and presents some elaborations and extensions to the algorithm.

```
Algorithm  OptAcc( $\langle \mathcal{F}, \mathcal{H}, \Upsilon_0 \rangle$, $\epsilon$, $\delta$ )
  • $\ell \leftarrow 0$      $k \leftarrow 1$
L1: Let $S \leftarrow \{\}$    Neigh $\leftarrow \{\, \tau_{i,j}(\Upsilon_\ell) \,\}_{i,j}$
L2: Get query/answer  $\langle q_k,\, a_k \rangle$  from oracle $\mathcal{O}$.
    Let  $S \;\leftarrow\; S \cup \{q_k\}$    $k \;\leftarrow\; k+1$
  • If there is some $\Upsilon' \in$ Neigh such that
        $c(\Upsilon',\, q_k) - c(\Upsilon_\ell,\, q_k) \;\geq\; \sqrt{2|S| \ln\left(\frac{k^2\ |\mathrm{Neigh}|\ \pi^2}{3\,\delta}\right)}$   (3)
    then let  $\Upsilon_{\ell+1} \leftarrow \Upsilon'$,  $\ell \leftarrow \ell + 1$.
    Return to L1.
  • If  $|S| \;\geq\; \frac{8}{\epsilon^2} \ln\left(\frac{k^2 |\mathrm{Neigh}| \pi^2}{3\ \delta}\right)$      and
        $\forall\, \Upsilon' \in$ Neigh,  $c(\Upsilon',\, q_k) - c(\Upsilon_\ell,\, q_k) \;\leq\; \frac{\epsilon\,|S|}{2}$,   (4)
    then halt and return as output $\Upsilon_\ell$.
  • Otherwise, return to L2.
```

Figure 1: Code for OptAcc

In more detail, OptAcc takes as arguments an initial representational system $\mathrm{R}_0 = \langle \mathcal{F}, \mathcal{H}, \Upsilon_0 \rangle$ along with parameters $\epsilon, \delta > 0$. It also uses $\mathcal{T} = \{\tau_{i,j}\}_{1 \leq i < j \leq n}$, a particular set of $O(n^2)$ possible transformations, 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 sequence to just before the i^{th} term. The set $\mathcal{T}[\Upsilon] = \{\, \tau_{i,j}(\Upsilon) \,\}_{i,j}$ define Υ's neighbors. OptAcc will climb from Υ_i to one of its neighbors, $\Upsilon' \in \mathcal{T}[\Upsilon]$, if this Υ' is statistically likely to be superior to Υ_i, based on the sequence of queries and answers produced by the oracle. This constitutes one hill-climbing step; in general, OptAcc will perform many such steps, climbing from Υ_0 to Υ_1 to Υ_2, etc., until terminating on reaching Υ_m.[8] Theorem 1 states our main theoretical results:

Theorem 1 *The* OptAcc($\langle \mathcal{F}, \mathcal{H}, \Upsilon_0 \rangle$, ϵ, δ) *algorithm incrementally produces a series of orderings* $\Upsilon_0, \Upsilon_1, \ldots, \Upsilon_m$ *(processing at most a polynomial number of samples at each stage) such that, with probability at least* $1 - \delta$*,*

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

The basic code for OptAcc appears in Figure 1.[9] In essence, OptAcc will climb from some Υ_j to a new $\Upsilon_{j+1} \in \mathcal{T}[\Upsilon_j]$ if Υ_{j+1} is likely to be strictly better than Υ_j; *i.e.*, if we are highly confident that

[7] All proofs appear in the expanded version [GS92].

[8] Actually, we only know that this algorithm will terminate with high probability; see Note2.

[9] That code uses "$c(\Upsilon_\alpha,\, q)$" to refer to "$c(\langle \mathcal{F},\, \mathcal{H},\, \Upsilon_\alpha \rangle,\, q)$".

$C[\Upsilon_{j+1}] > C[\Upsilon_j]$. As each query q_i is selected independently according to some fixed distribution, Chernoff bounds [Che52, Bol85] show that the observed sample average, $\frac{1}{k}\sum_{i=1}^{k} c(\Upsilon_\alpha, q_i)$, converges exponentially fast to the population mean, $C[\Upsilon_\alpha]$.

The OptAcc algorithm uses these bounds to determine both how confident we should be that $C[\Upsilon'] > C[\Upsilon_\ell]$ (Equation 3) and whether any "$\mathcal{T}$-neighbor" of Υ_ℓ (*i.e.*, any $\tau_{ij}(\Upsilon_\ell)$) is more than ϵ better than Υ_ℓ (Equation 4). The extended paper [GS92] discusses how to compute the values of $\sum_{q \in S} c(\tau_{ij}(\Upsilon_\ell),\, q) - c(\Upsilon_\ell,\, q)$ for each $\tau_{ij\ell} \in \mathcal{T}$.

Note1. Our descriptions 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 many situations, we may want to consider each hypothesis to be the conjunction of a *set* of sub-hypotheses, which must all collectively be asserted to reach a conclusion. Here, we can view $\mathcal{H} = \mathcal{P}[H]$ as the power set of some set of "sub-hypotheses", H.

Note2. As this OptAcc process can require general theorem proving (*e.g.*, to determine whether $\mathcal{F} \cup \{h_i\} \models^? q/\mathcal{O}'[q]$), it will in general be computationally intractable. However, if this process is decidable (*e.g.*, if we are dealing with propositional theories), then OptAcc will terminate with probability 1, as the space of strategies is finite; see [CG91]. Finally, each iteration in the OptAcc algorithm is polytime if the $\mathcal{F} \cup \{h_i\} \models^? q_k$ computation is polytime; *e.g.*, if we are dealing with propositional Horn theories or propositional 2-CNF, etc.

Note3. If we set $\epsilon = 0$, the OptAcc$(R_0, 0, \delta)$ process will not terminate. Its behavior is still quite useful: it will, with high probability, produce a series of better and better strategies. Indeed, we can view this system as an *anytime algorithm* [DB88] as, at any time, it will return a workable result (here, the ordering produced at the j^{th} iteration, Υ_j), with the property that the longer we wait, the better the result.

Note4. Recall that, in general, we need to compute the values of $\sum_{q \in S} c(\tau_{ij}(\Upsilon_\ell),\, q) - c(\Upsilon_\ell,\, q)$ for each τ_{ij} in $\mathcal{T}$. Above, we obtained this information by determining whether $\mathcal{F} \cup \{h_i\} \models^? q/\mathcal{O}'[q]$ 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 [Gre92] 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.

Note5. This paper considers only one type of transformation to convert one representational system into another — *viz.*, by rearranging the set of hypotheses. There are many other approaches, *e.g.*, by eliminating some inappropriate sets of hypotheses [Coh90, Won91], or by 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 representational systems. We can then consider the same objective described above: to identify which element has the highest expected accuracy.

Here, as above, the expected accuracy 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 the one shown above) 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 representational systems, etc.)

Note6. There are several obvious extension to this work: First, we have so far insisted that each answer to a query is 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 `$10,000` is more precise than knowing only that `watch`$_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 -1, 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 $c(R, q)$ 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 $c(R, q)$ function.

Notice finally that we have completely discounted the computational cost associated with arriving at the answer. Within this framework, we can consider yet more general $c(\cdot, \cdot)$ functions, that can even incorporate the user's tradeoffs between accuracy and efficiency, etc. 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]. 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 representational 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, add to the hypothesis set an additional hypothesis h_3 : $\mathtt{N}_{AE}(x)$ and to the fact set the rule $\forall x.\ \mathtt{A}(x)\ \&\ \mathtt{E}(x)\ \&\ \mathtt{N}_{AE}(x) \Rightarrow \mathtt{S}(x,\ \mathtt{IDK})$. Here, a representational system that accepts the $\mathtt{N}_{AE}(\mathtt{Z}_{15})$ hypothesis will produce the answer IDK to the query $\mathtt{S}(Z_{15},\ y)$.

Note7. The motivation underlying this work is similar to the research of [Sha89] and others, who also use probabilistic information to order the various default rules. Our work differs by providing a way of obtaining the relevant statistics, rather than assume that they are known *a priori*, or can be computed purely from static analysis of ground facts in the database.

4 Conclusion

Many nonmonotonic reasoning systems are ambiguous, in that they can produce many individually plausible but collectively incompatible solutions to certain queries. Unfortunately (at most) one of these solutions is correct; the challenge then is to determine which one. 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 preferring the credulous system that is correct most often — *i.e.*, that 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 have defined a learning algorithm, OptAcc, that sidesteps these problems by using a set of samples (each consisting of a query and its correct solution) to obtain an estimate of the unknown distribution, and by using a particular 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 it requires only a polynomial number of samples to climb from one credulous system to another.

References

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

[Bre89] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *IJCAI-89*, 1989.

[CG91] W. Cohen and R. Greiner. Probabilistic hill climbing. In *CLNL-91*, 1991.

[Che52] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sums of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

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

[DB88] T. Dean and M. Boddy. An analysis of time-dependent planning. In *AAAI-88*, 1988.

[GE91] R. Greiner and C. Elkan. Measuring and improving the effectiveness of representations. In *IJCAI-91*, 1991.

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

[GO91] R. Greiner and P. Orponen. Probably approximately optimal derivation strategies. *KR-91*, 1991.

[Gre92] R. Greiner. Learning near optimal horn approximations. In *Knowledge Assimilation Symposium*, 1992.

[Gro91] B. Grosof. Generalizing prioritization. In *KR-91*, 1991.

[GS92] R. Greiner and D. Schuurmans. Producing more accurate representational systems. TR, Siemens Corporate Research, 1992.

[Hau88] D. Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 1988.

[Mit80] T. Mitchell. The need for bias in learning generalizations. TR CBM-TR-117, 1980.

[OG90] P. Orponen and R. Greiner. On the sample complexity of finding good search strategies. In *COLT-90*, 1990.

[OM90] D. Ourston and R. Mooney. Changing the rules: A comprehensive approach to theory refinement. TR, Dept of Computer Science, University of Texas, 1990.

[PGA86] D. Poole, R. Goebel, and R. Aleliunas. Theorist: A logical reasoning system for default and diagnosis. TR CS-86-06, 1986.

[Rei87] R. Reiter. Nonmonotonic reasoning. In *Annual Review of Computing Sciences*, volume 2, 1987.

[RG87] S. Russell and Benjamin N. Grosof. A declarative approach to bias in concept learning. In *AAAI-87*, 1987.

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

[SK75] H. Simon and J. Kadane. Optimal problem-solving search: All-or-none solutions. *Artificial Intelligence*, 6:235–247, 1975.

[Smi89] D. Smith. Controlling backward inference. *Artificial Intelligence*, 39(2):145–208, June 1989.

[vA90] P. van Arragon. Nested default reasoning with priority levels. In *CSCSI-90*, 1990.

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

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