Knowledge Infusion*

Leslie G. Valiant
Division of Engineering and Applied Sciences
Harvard University

Abstract

The question of how machines can be endowed with the ability to acquire and robustly manipulate commonsense knowledge is a fundamental scientific problem. Here we formulate an approach to this problem that we call knowledge infusion. We argue that robust logic offers an appropriate semantics for this endeavor because it supports provably efficient algorithms for a basic set of necessary learning and reasoning tasks. We observe that multiple concepts can be learned simultaneously from a common data set in a data efficient manner. We also point out that the preparation of appropriate teaching materials for training systems constructed according to these principles raises new challenges.

Introduction

Perhaps the most important advance that might be made to make computers more useful to humans is that of empowering them with the ability to acquire and manipulate commonsense or unaxiomatized knowledge. Knowledge infusion is a particular approach to handling unaxiomatized knowledge. We define it here to mean any process of knowledge acquisition by a computer that satisfies the following three properties:

1. The stored knowledge is in such a form that principled reasoning on it can be carried out computationally feasibly. In particular two or more pieces of knowledge can be chained together so as to derive a conclusion, and the confidence in the conclusion that is justified can be estimated in a principled way.
2. The stored knowledge and principled reasoning are such that the reasoning is robust to errors in the inputs to the system, to uncertainty in the knowledge, or to the gradual changes in the truth of various pieces of knowledge. Robustness is ensured and brittleness avoided by means of a large-scale continuous process of learning from an environment.
3. The knowledge acquisition can be done automatically on a massive scale, and simultaneously for many concepts, being permitted both by the computational efficiency of the algorithms as well as by their economy in the use of external data.

In this paper we propose a theoretical basis for knowledge infusion in terms of robust logic. Knowledge infusion based on robust logic can be distinguished from other current approaches to reconciling learning and reasoning, in three important respects. First it provides a *common semantics* for both the learning and the reasoning. Second it places at the forefront the criterion of the existence of efficient, in particular polynomial time and data efficient, algorithms for all aspects of the process, including learning, reasoning and dealing with multiple objects and relations. Third, its goal is quite ambitious - in contrast with the term "knowledge extraction" as widely used - its aim includes the extraction from data of knowledge that will be useful in contexts not foreseen at the time of the extraction.

In this paper we discuss five issues.

First, we describe the algorithmic approach and point out that the robust logic is able to address the necessary challenges because it has an appropriate semantics, and also because it incorporates a quantitative notion of working memory.

Second, we observe that beside algorithmic issues there is the additional problem of preparing data or, in our terminology, teaching materials, for such systems.

Third, in knowledge infusion there is no one fixed target for the learning - one needs to attempt to learn as many targets as possible from the available data. The question arises as to the needed sample complexity of simultaneously learning many concepts from the same data set.

Fourth, if we are assuming that the source of knowledge is data, the question arises as to why we need reasoning at all. Can answers to all queries be derived equally well by memorizing all the data, and answering each query as it arises by analyzing the data anew, as suggested in Khardon and Roth (1997)?

Lastly, for completeness, we briefly describe a simplified robust logic on which such systems might be based.

Knowledge Infusion

Knowledge infusion was first outlined as a goal in (Valiant 2000a) where an architecture for systems for achieving it was described. It was soon realized that some more foun-

*This work was supported by grants NSF-CCR-03-10882, NSF-CCF-0432037 and NSF-CCF-04-27129.

dational work was required to define a semantics for knowledge infusion, and (Valiant 2000b) took a first step in that direction by introducing *robust logics*. In particular, the latter paper described polynomial time algorithms for both the learning and reasoning processes there described, and, perhaps most importantly, a common semantic basis for both.

The philosophical approach we use can be summarized very simply as follows: *We adapt the semantics of learning so that it also applies to the reasoning problem. Good empirical performance on previously unseen examples is the accepted criterion of success in supervised learning. It is exactly this criterion that we believe needs to be achieved for reasoning systems to be viewed as robust.*

The most immediate theoretical problems that have to be addressed are:

1. Knowledge infusion has both a logical aspect needed for reasoning, as well as a statistical learning aspect needed to ensure that the knowledge base is maintained robustly so as to resist errors and inconsistencies. A common semantics is needed that works for both aspects.
2. A theory of knowledge infusion has to provide efficient computational solutions in an area where there are at least three separate identifiable sources of potential exponential complexity: learning (even in a propositional setting), reasoning (even in a propositional setting), and handling multi-object or relational representations.

In order to address (1) we have rules that may be chained as logical rules, but their semantics is that of learning. A rule, which may have been either preprogrammed or learned, is not assumed to be true in any logical sense. Instead it is constantly tested against examples observed in the world and the degree of belief that is justified in it depends empirically on how often it is found to be true.

In order to address (2) we have to have a learning framework that allows relational expressions but is nevertheless polynomial time bounded. Learning in a relational setting has been explored in a variety of contexts (Haussler, 1989; Quinlan, 1990; Lavrac *et al.* 1991; Kramer *et al.* 2001; Cumby and Roth, 2002, 2003; Valiant, 1985.) It clearly offers some challenges, both computational and conceptual, beyond those of propositional learning.

The proposed formalism is based on Valiant (2000b) and is, very briefly, of the following form. The rules learned are equivalences. The right hand sides are simple predicates. The left hand sides may be complex conditions, typically a linear inequality in terms of features that are themselves conjunctions of basic relational predicates with local quantifications. The rules, though equalities, may be chained together in reasoning. Polynomial time algorithms exist for both learning and sound and complete reasoning, in senses there defined, provided the base relations have constant arities.

An important aspect of this robust logic is that it adapts the notion of "working memory" from cognitive science (Miller, 1956; Newell 1987; Miyake and Shah, 1999) into mathematical logic and learning theory. In this new formalization, existential and universal quantifications are over objects in the working memory, and not an unlimited and ill-specified world. Using learning algorithms one aims to learn rules that are highly reliable on the distribution of the contents of the working memory, which we call *scenes*.

Working memory can be viewed as having two important roles. First, it makes the computations internal to a cognitive system possible. This role may be thought of as loosely analogous to that of registers in computer hardware, in providing rich computational capabilities on a limited amount of data locally brought together, capabilities which would be too onerous to be feasible without this limitation. Second, it restricts and localizes the system's view of the outside world to a window that is small enough that it provides a domain that is learnable. It is this second role that we emphasize here. In robust logic this second role is exploited to provde for algorithmic efficiency in learning and reasoning, independent of any assumptions about how the first role might be implemented in cognition.

Teaching Materials

The first task, the one we emphasize in this paper, is that of understanding the algorithmic possibilities and limitations inherent to knowledge infusion. There is, however, the additional problem of understanding how best to create *teaching materials* for such systems. This choice of terminology is made in acknowledgement of the fact that enormous efforts are expended by humans in preparing such materials for every stage of human education. We interpret this to mean that even if a system has viable algorithms for knowledge infusion, as presumably humans have, there is still the need for carefully prepared teaching materials. Hence, we believe that even when the algorithmic issues are well understood, substantial problems may remain with regard to creating teaching materials. In particular, one expects that the content of such materials will need to be organized into layers. The system may need to learn one layer well before it can learn the next layer.

Learning Many Concepts Simultaneously From Few Examples

When presented with a scene it is legitimate to view it as an example of every concept that occurs in it, and a counterexample of every concept from a given vocabulary that does not. Knowledge infusion therefore raises fundamental issues regarding the efficiency with which knowledge can be extracted from data, in the case that the same piece of data is to be used to learn many concepts simultaneously.

One basic question is whether when one learns N distinct functions, rather than just one, does one need more, say N times as many, examples? The fortuitous answer is that the number of examples needed increases little with the number of target functions.

In computational learning theory the question of how much time and how many examples are needed for learning a single propositional concept from a concept class has been studied extensively (Valiant, 1984; Blumer et al. 1989; Ehrenfeucht *et al*. 1989). The definitions and some simple results can be easily adapted to the problem of learning many concepts simultaneously.

Definition For a polynomial $m(n, \epsilon, \delta)$ in arguments $n, 1/\varepsilon, 1/\delta$, algorithm L is an $m(n, \varepsilon, \delta)$-PAC *learning algorithm* for concept class C iff for any n, ε, and δ, any distribution D of inputs over n variables, and any function $c \in C$ with n variables, after seeing $m(n, \varepsilon, \delta)$ random examples from D, L will with probability at least 1 -δ output a hypothesis for c of accuracy at least 1- ε. Also, L is an $m(n, \varepsilon, \delta, N) - simultaneous - PAC$ *learning algorithm for C* if for any D after seeing $m(n, \varepsilon, \delta, N)$ random examples, it can with probability at least 1 - δ learn a set of N hypotheses for N arbitrary concepts $c_1, \cdots, c_N \in C$, each one accurate to at least 1- ε.

Proposition *If there is an $m(n, \epsilon, \delta)$-PAC learning algorithm L for concept class C then there is an $m(n, \epsilon, \delta^*, N)$-simultaneous-PAC learning algorithm for C, where $m(n, \varepsilon, \delta^*, N) = m(n, \varepsilon, \delta^*/N)$.*

Proof We take a sample of $m(n, \varepsilon, \delta)$ random examples from D and on this common set apply L to learn each of the N target functions in turn. Then the probability of obtaining a hypothesis with error greater than ε is at most δ for each one, and the probability of getting such an error in any one of the N hypotheses is, by the union bound, at most $N\delta$. Substituting $\delta^* = N\delta$ gives the result. ■

This observation is already very encouraging since it says that the cost of simultaneous learning of N concepts to confidence δ^* is simply that of learning a single concept to a higher confidence of δ^*/N. In the well-known bounds on sample complexity this requires a multiplicative increase of at most $\log N$, and often much less. This is illustrated by the following case of attribute efficient learning. Littlestone (1988) showed that for learning disjunctions of at most k variables out of a set of n variables there is a polynomial time $O(k \log n)$ mistake bounded algorithm. From this it follows by applying a general conversion to PAC learning (Littlestone, 1989; Schurmans and Greiner, 1995) that there is a $O((1/\varepsilon)(k \log n + \log(1/\delta)))$-PAC learning algorithm. Hence, we can deduce the corollary that:

Corollary *Disjunctions of at most k variables out of a set of n variables have a $O((1/\varepsilon)(k \log n + \log(N/\delta)))$-simultaneous-PAC learning algorithm.*

In the standard setting Littlestone's result has the consequence that when we are to learn a function of n variables the sample size dependence on n is only logarithmic, as opposed to the general linear lower bound (Ehrenfeucht, *et al.* 1989). The restriction needed is that each concept depends on only a few, namely k, of the n variables. This k appears as a factor in the bound, but provides a great saving if it is small, constant or logarithmic, say. The simultaneous version expressed by the corollary above states the following: when N concepts are to be learned simultaneously, the number of examples needed goes up by at most a factor of two if $N < n^k$, and by at most an additive $(\log N)/\varepsilon$ in general. We note that Littlestone's result, is more general than stated here, and applies not just to disjunctions but to certain more general linear separators. The PAC translations for that class convert to simultaneous learning results in a similar way.

The phenomenon that attribute-efficient learning is possible even in the simultaneous learning setting we regard as an essential prerequisite for cognitive computations. It is also fundamental to our implementation of knowledge infusion, in which variables proliferate according to general combinatorial schemas.

While the simple arguments above are satisfying, we go on here to spell out what happens in the more general case that some of the concepts we attempt to learn do not lie in the class of concepts that the learning algorithm at hand is able to learn. This addresses the practical problem that among the many concepts that may be targets, some may be too complex for the algorithm with respect to the features, while others may be learnable. We formulate a Simultaneous Occam Theorem for this general case. In particular, we shall observe, that the now classical Occam bounds (Blumer, *et al.* 1987) on sample size hold unchanged for any value N. *Thus, to this level of analysis, we are at liberty to reuse the same sample set to learn any number of concepts.* This observation provides some explanation, we believe, of how it is possible, for humans to learn apparently so much from experiences that are relatively so few though possibly complex.

In formulating this result we will go the further step of considering error measures that are appropriate when the function learned is unbalanced. First we note that the simplest way of measuring the error of a hypothesis h is to say Error(h) = Prob[$h(x) \neq f(x)$], where f is the true function that is being learned, and probabilities are taken over random draws of an example x from distribution D. If f is not balanced, so that it is true very rarely, then it is useful to measure error using:

$$\text{Precision}(h) = \text{Prob}[h(x) = f(x) = 1]/\text{Prob}[h(x) = 1],$$
$$\text{Recall}(h) = \text{Prob}[h(x) = f(x) = 1]/\text{Prob}[f(x) = 1],$$

and to define Precision Error(h) $= 1 -$ Precision(h), and Recall Error(h) $= 1 -$ Recall(h).

Simultaneous Occam Theorem *Suppose examples are drawn from a domain X_n over variables $x_1, \cdots, x_n$ according to a fixed unknown distribution D_n. Suppose that we have a list $F = \{f_1, \cdots, f_N\}$ of target functions to learn where each one classifies some function $f_k(Y_k)$ according to the values of Y_k where Y_k is a subset of $\{x_1, \cdots, x_n\}$. Suppose that A is a learning algorithm that when given labeled examples from X_n returns hypotheses $h_1, \cdots, h_N$ from a class H_n for the functions in F. Suppose that in one experiment a set S of m randomly and independently chosen examples from X_n was drawn, and on this set the learning algorithm A returned hypotheses for F such that for some nonempty subset H^* of these hypotheses, all the hypotheses in H^* predict all the sample points in S correctly. Then for any $m > (1/(2\varepsilon))(\log_2 |H_n| + \log_2(1/\delta))$ and any $\varepsilon < \frac{1}{2}$, if, for every $h_i \in H^*$, $\varphi_i = Prob[h_i(x) = 1]$,*

(i) Prob[There is some $h_i \in H^*$ with error$(h_i) > \varepsilon] < \delta$,
(ii) Prob[There is some $h_i \in H^*$ with precision error$(h_i) > \epsilon/\varphi_i] < \delta$, and

(iii) Prob[There is some $h_i \in H^*$ with recall error$(h_i) > \varepsilon/(\varphi_i - \varepsilon)] < \delta$.

Proof The probability that any fixed function $h \in H_n$ that has error greater than ε agrees with all m examples in a set S drawn randomly and independently from X_n according to probability distribution D_n is less than $(1-\varepsilon)^m$. Since there are $|H_n|$ such possible functions it follows that the probability that even one of the possibly many functions $h \in H_n$ that have error greater than ε agrees with S is, by the union bound, less than $|H_n|(1-\varepsilon)^m$. But this last quantity is clearly upper bounded by δ if $(1/\delta)|H_n|(1-\varepsilon)^m < 1$. Taking natural logarithms gives

$$\log_e(1/\delta) + \log_e|H_n| + m\log_e(1-\varepsilon) < 0.$$

Then the required

$$m > (1/(2\varepsilon))[\log_2|H_n| + \log_2(1/\delta)]$$

follows since - $\log_e(1-\varepsilon) = \varepsilon + \varepsilon^2/2 + \varepsilon^3/3 + \cdots < (2\log_e 2)\varepsilon$ if $\varepsilon < \frac{1}{2}$.

By definition,

$$\text{Error}(h) = \text{Prob}[h(x) = 1 \,\&\, f(x) = 0] + \text{Prob}[h(x) = 0 \,\&\, f(x) = 1].$$

Also Precision(h) equals

$$\begin{aligned}&\text{Prob}[h(x) = f(x) = 1]/\text{Prob}[h(x) = 1]\\ &= (\text{Prob}[h(x) = 1] - \text{Prob}[h(x) = 1 \& f(x) = 0])/\text{Prob}[h(x) = 1]\\ &= 1 - \text{Prob}[h(x) = 1 \& f(x) = 0]/\text{Prob}[h(x) = 1]\\ &\geq 1 - \varepsilon/\varphi,\end{aligned}$$

and result (ii) follows, for otherwise (i) would be contradicted.

Finally Recall(h) equals

$$\begin{aligned}&\text{Prob}[h(x) = f(x) = 1]/\text{Prob}[f(x) = 1]\\ &= (\text{Prob}[f(x) = 1] - \text{Prob}[f(x) = 1 \& h(x) = 0])/\text{Prob}[f(x) = 1]\\ &= 1 - \text{Prob}[f(x) = 1 \& h(x) = 0]/\text{Prob}[f(x) = 1]\\ &\geq 1 - \varepsilon/(\text{Prob}[h(x) = 1] - \text{Prob}[f(x) = 0 \& h(x) = 1])\\ &\geq 1 - \varepsilon/(\varphi - \varepsilon),\end{aligned}$$

and result (iii) follows. ■

We note that there are no assumptions on the source of data except that they are drawn from a fixed distribution. Surprisingly, this bound on m does not depend on the size of H^*. Consider, for example, the case that $(1/\varepsilon), \log|H_n|$ and $\log(1/\delta)$ and hence m, are all polynomially bounded in terms of the number of variables n, but F is exponentially large in terms of n. Suppose that after a run of the algorithm an exponentially large subset H^* of predicted hypotheses turn out to agree exactly with the sample. *Then we are justified in having high confidence $1-\delta$ in every single one of the even exponentially many hypotheses that are found to agree with the training set.* This applies also to parts (ii) and (iii) which require estimates of the fraction of the distribution on which the various $h_i \in H^*$ have value 1.

We interpret the above observations as strong evidence that knowledge infusion does not require an inordinate amount of data.

Are Rules Superfluous?

We now discuss the following fundamental question: If we have access to observations why can we not make predictions directly from the observations rather than first learn rules and then make deductions. For example, learning $A \Rightarrow B$ and $B \Rightarrow C$ from examples, and then chaining these two rules together on an instance for which A holds is superfluous if the data that provides overwhelming evidence for $A \Rightarrow B$ and $B \Rightarrow C$ both holding also provides the same evidence for $A \Rightarrow C$ holding. Khardon and Roth (1997) offer a theory of "learning to reason," in which reasoning is made redundant in some such sense. Here we discuss three justifications for the value of rules in the current context.

First, we illustrate one aspect of the probabilistic semantics of reasoning in robust logic. Consider the two simple rules:

$$A \vee \cdots \equiv B \tag{1}$$

and

$$(B \wedge C) \vee \cdots \equiv D. \tag{2}$$

If these have been learned to accuracy 99% then that means that on a random natural scene drawn from the given distribution the probability that one side is satisfied but not the other is less than 1%. The deduction process of the robust logic will chain these rules together to deduce that for a random scene from the same distribution for which $A \wedge C$ holds, D will hold also. This is justified in the sense that even if the two rules are not independent but arbitrarily correlated, it follows from a simple union bound argument that the conclusion must hold with error at most the sum of the errors of the two rules, namely 2%.

A robust logic based system can be viewed as performing a thought experiment when chaining. It is not given a complete scene but one where some of the features have been "obscured" and it has to deduce what these values are likely to be in natural cases. Suppose that B holds for 10% of the distribution, and of these instances a half are accounted for by A being also true (i.e. A holds for 5% of the distribution.) Suppose that D holds for 18% of the distribution, and of this a third (6% of the distribution) is accounted for by $B \wedge C$. In this case the system will make the deduction that D holds, if both A and C hold, even though the fraction of the distribution on which A and C both hold may be arbitrarily small, and too small for any direct observations as to whether D holds for those instances.

The above example shows that deduction in robust logic has the following semantics: For situations that are too rare to observe the deduction will effectively assume that the relevant rules are probabilistically independent, but for situations that are frequent enough to observe, the logic will protect against invalid conclusions. In the case of the above example, if on 3% of examples A and C both hold but D is false, then at least one of the rules (1) or (2) has error greater than 1%, and it would have been erroneous to learn it.

With this backdrop, we now list three reasons for why reasoning is useful in our setting, even when learning is available.

First, as we illustrated in the example, while the robust logic protects against drawing false conclusions in situations in which there is significant statistical evidence to the contrary, it also chains rules to derive conclusions in situations that are too infrequent for the statistics of the data to strongly support a deduction. In those cases the rules effectively invoke some independence assumptions, in an arguably plausible manner.

Second, the robust logic learns equivalence rules of certain forms. In any resulting deduction, some rule equivalent to chaining several of the simple rules that have been learned may be implicitly invoked. In the previous paragraph we pointed out that this implicit rule may not be statistically supported by the data. Another possibility is that it is too complex to be predicted directly from the data using a given algorithm.

Third, if programmed rules are allowed as inputs to the system in addition to the training data, then reasoning cannot be immediately supplanted by learning. For example, in a natural language setting useful equivalents of programmed rules are dictionaries, of synonyms or antonyms, for example. There is the issue of how the programmed rules are to be interpreted in the robust logic. In general they can be updated in the same way as learned rules. However, in the case of dictionaries, for example, a simple progmatic approach is to regard them as always true and not subject to updating by learning. Then in learning other rules, instead of learning from the input scenes directly one can augment the scenes by deriving all consequences of them using the programmed rules, and regard the distribution of augmented scenes so produced as the training set for the learned rules.

Finally, we note that while the strength of the robust logic described is that it has a principled semantics, we do not advocate that pragmatism be avoided altogether when building systems based on it. For example, deductions about a predicate typicaly will be made on scenes where the truth-value of the predicate is unknown, rather than known but obscured.

A Simplified Robust Logic

For completeness we describe here a simplified of robust logic that, we believe, is a good basis for knowledge infusion systems.

We consider a fixed set $\mathcal{R}$ of t *base relations* R_i of arities $\alpha(i)$ respectively, where all the $\alpha(i) \leq \alpha$. A *scene* consists of n generic token objects $\mathcal{A} = \{a_1, \cdots, a_n\}$, and a vector L that for all $R_i \in \mathcal{R}$ and all $n^{\alpha(i)}$ substitutions of token objects for the $\alpha(i)$ arguments of R_i states whether R_i holds for that substitution. The components of L have values "1" or "0" indicating the case of R_i for each combination of objects, *e.g.*, one component of L may indicate $R_7(a_2, a_3, a_8) = 0$.

The examples input to our learning and reasoning systems will be such scenes, each derived, for example, from a natural language sentence or paragraph. Before we proceed with the rest of the formalism we need to explain our treatment of incomplete information in this simplified system:

We shall interpret a value 1 in L as "determined to be true" and a value 0 as "unspecified or determined to be false." The learning algorithm will therefore take inputs of this form and make predictions according to these two categories. Note that if for each relation we add a second relation that represents its contrary then we can express the "determined to be false" case also. Then, in effect, a three-valued logic will be realized using two binary variables for each relation.

When analyzing reasoning we use the further notion of partial information, that of *obscuring*. An obscured scene is a scene in which some of the components of L have been replaced by the third value "obscured". When we test the success of a deduction algorithm for making correct predictions on a relation R_i, we obscure a possibly known component of L, such as $R_7(a_2, a_3, a_8)$, and see whether that obscured information can be accurately predicted by the deduction process applied to the other components of L. We note that in standard PAC learning the target variable to be predicted can be viewed as obscured in the training process in exactly this sense.

Let D be a distribution on the vectors L over such scenes. It corresponds to the distribution of natural examples that arise from the world as perceived in the mind's eye of one agent. In this simplified RL we assume here that D is symmetric under permutations of the token objects, reflecting the total absence of structure among the tokens.

A *rule* q is of the form $\forall x_1, \forall x_2, \cdots, \forall x_{\alpha(i)}$ [lhs $\equiv$ rhs] where rhs is of the form $R_i(x_1, x_2, \cdots, x_{\alpha(i)})$ and lhs has the same set $x_1, x_2, \cdots, x_{\alpha(i)}$ of free variables and is some expression q from some permitted universe E of expressions involving $\mathcal{R}$. We call the class of such rules $(E, \mathcal{R})$-*rules*.

An *independently quantified expression*, or IQE, for a relation T, of arity three for example, is of the form $e = \Delta x \Delta y \Delta z T(x, y, z)$ where each occurrence of Δ may be either an existential quantifier $\exists$, or a null quantifier. Further, the non-null quantifiers, here the existential ones, are totally independent of any other IQE. In other words, those of the variables x, y, z that have an existential quantifier are local to just the one relation T. The remaining variables are the free variables. Both bound and free variables refer only to tokens in the scene. For any one unobscured scene, any IQE such as $e = \exists x \exists z T(x, y, z)$, and any binding of the free variables of e, in this case y, to the objects in the scene, a Boolean truth value is determined therefore.

Now given a set of base relations $\mathcal{R}$ we shall define a *schema of* IQEs as follows: We consider a bounded number of ways of conjoining members of $\mathcal{R}$ each with some quantification, but with a common set of free variables. For example

$$\{R_1(x), \exists y R_2(x, y), \exists y R_3(y) R_4(x, y), \\ \exists y \exists z R_5(x) R_6(y) R_7(x, y, z)\}$$

is such a schema having four forms, with the common free variable x. A *schema of* IQEs for $\mathcal{R}$ is then the set of such quantified conjunctions with all possible combinations of substitutions of members of $\mathcal{R}$ with the appropriate arities as specifed by the forms allowed in the schema. Each form specifies a combination of the syntax allowed for a T and the syntax of its quantifiers.

The class of $(E, \mathcal{R})$-rules that will form the basis of our RL can now be defined. To obtain E from a base set of relations $\mathcal{R}$ we generate all possible members of a specific schema of IQEs (e.g. schema $\{R_1(x), \exists y R_2(x,y), \exists y R_3(y) R_4(x,y)\}$) and let E be a class of Boolean functions that is easily learned and has this set of IQE's as its Boolean variables.. A very convenient choice of learning algorithm is Littlestone's Winnow (Littlestone, 1987) since this has provably good learning properties and performs well in practice on natural data sets (Golding and Roth, 1999). In that case E will be linear threshold functions. In general, if we apply a propositional learning algorithm to the Boolean vector of values of all possible IQEs of schema, then we can exploit the propositional learning algorithm in the normal way, and all its properties, such as sample bounds, attribute-efficiency and resilience to errors will be inherited in the relational domain.

Thus rules of the specified kind can be learned that are accurate in the PAC sense on the distribution of possible scenes. It is shown in (Valiant 2000b) that these rules can be chained to derive deductions about individual scenes. The chaining process is sound and complete in a defined sense. For efficient computaition one needs that all the base relations have bounded arities, say $\alpha \leq 3$. Given this one assumption all the learning and reasoning processes require only polynomially bounded resources in terms of the number of tokens, the number of base relations, and the inverse of the errors that are to be tolerated.

The simplified robust logic described here can be extended in various natural ways. The treatment in (Valiant, 2000b) can be viewed as an extension in the following four directions: preconditions are allowed for each rule, the distribution D need not be symmetric, bindings need not be one-to-one, and universal quantifiers are allowed in addition to existential quantifiers.

Acknowledgement

It is a pleasure to thank Loizos Michael for numerous conversations.

References

Blumer, A.; Ehrenfeucht, A.; Haussler, D.; and Warmth, M. 1987. Occam's razor. *Inf. Process. Lett.* 24(6):377–380.

Blumer, A.; Ehrenfeucht, A.; Haussler, D.; and Warmuth, M. 1989. Learnability and the Vapnik-Chervonenkis dimension. *JACM* 36(4):929–965.

Comby, C., and Roth, D. 2003. On kernel methods for relational learning. *ICML* 107–114.

Cumby, C., and Roth, D. 2002. Learning with feature description logics. In *Proc. Int. Conf. on Inductive Logic Programming*. 32-47.

Ehrenfeucht, A.; Haussler, D.; Kearns, M.; and Valiant, L. 1989. A general lower bound on the number of examples needed for learning. *Inf. Comput.* 3:247–261.

Golding, A., and Roth, D. 1999. A winnow-based approach to context-sensitive spelling corrections. *Machine Learning* 34(1-3):107–130.

Haussler, D. 1989. Learning conjunctive concepts in structural domains. *Machine Learning* 4:7–40.

Kearns, M., and Valiant, L. 1994. Cryptographic limitations on learning boolean formulae and finite automata. *JACM* 1:67–95.

Khardon, R., and Roth, D. 1997. Learning to reason. *JACM* 697–725.

Khardon, R.; Roth, D.; and Valiant, L. 1999. Relational learning for NLP using linear threshold elements. *IJCAI99* 911–917.

Kramer, S.; Lavrac, N.; and Flach, P. 2001. Propositionalization approaches to relational data mining. In Dzeroski, S., and Lavrac, N., eds., *Relational Data Mining*, 262–291. Springer-Verlag.

Lavrac, N.; Dzeroski, S.; and Grobelnik, M. 1991. Learning nonrecursive definitions of relations with linus. In Kodratoff, Y., ed., *Proceedings of the 5th European Working Session on Learning*, volume 482, 265–281. *Lecture Notes in Artificial Intelligence,* Springer-Verlag.

Littlestone, N. 1988. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *JACM* 2(4):2850–318.

Littlestone, N. 1989. From on-line to batch learning. In *Proc. COLT89*, 269–284. Morgan Kaufmann.

Miller, G. 1956. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psych. Rev* 63:81–97.

Miyake, A., and Shah, P., eds. 1999. *Models of Working Memory: Mechamisms of active maintenace and executive control*. New York, NY: Cambridge University Press.

Newell, A. 1987. *Unified Theories of Cognition*. Cambridge, MA: Havard University Press.

Pitt, L., and Valiant, L. 1988. Computational limitations on learning from examples. *JACM* 35:965–984.

Quinlan, J. 1990. Learning logical definitions from relations. *Machine Learning* 5:239–266.

Roth, D., and Yih, W. 2001. Relational learning via propositional algorithms: An information extraction case study. *IJCAI'01* 1257–1263.

Schuurmans, D., and Greiner, R. 1995. Sequential PAC learning. In *Proc. 8th COLT*, 377–384. ACM Press.

Terwijn, S. 2005. Probabilistic logic and induction. *J. of Logic and Computation* 15(4):507–515.

Valiant, L. 1984. A theory of the learnable. *CACM* 27(11):1134–1142.

Valiant, L. 1985. Learning disjunctions of conjunctions. In *Proc. 9th IJCAI*, 560–566. Morgan Kaufmann.

Valiant, L. 2000a. A neuroidal architecture for cognitive computation. *JACM* 47(5):854–882.

Valiant, L. 2000b. Robust logics. In *Artificial Intelligence Journal*, 642–651. 117 (2000) 231-353.