

Implicit Learning of Common Sense for Reasoning

Brendan Juba*
 Harvard University
bjuba@alum.mit.edu

Abstract

We consider the problem of how enormous databases of “common sense” knowledge can be both learned and utilized in reasoning in a computationally efficient manner. We propose that this is possible if the learning only occurs *implicitly*, i.e., without generating an explicit representation. We show that it is feasible to invoke such implicitly learned knowledge in essentially all natural tractable reasoning problems. This implicit learning also turns out to be provably robust to occasional counterexamples, as appropriate for such common sense knowledge.

1 Introduction

The means of acquisition and use of common sense knowledge is a central question of Artificial Intelligence, starting with McCarthy’s work [1959]. As a starting point, it is helpful to distinguish between common sense knowledge that is *learned* from typical experiences, and knowledge that is (at least in humans and many other animals) *innate*, specifically the *core knowledge* described by Spelke and Kinzler [2007]. We will only seek to address the learned knowledge here. Valiant [2006] has proposed that such knowledge can be formally captured by a logical syntax with a semantics derived from machine learning, *PAC-Semantics*. In particular, Valiant observed that the number of rules that can be soundly learned from a given data set is exponentially large in the amount of data provided. This efficiency of the data requirements is encouraging given that existing KBs of common sense knowledge such as CYC [Lenat, 1995] and Open Mind [Stork, 1999] contain millions of rules at present and continue to grow. Here, we will seek to address a puzzle left untouched by Valiant’s work, namely, *how is it that a computationally limited agent can so efficiently acquire and marshal such vast stores of knowledge?* We will argue that this is possible when the learning occurs only *implicitly*, i.e., without producing explicit representations of the learned knowledge. We will show specifically how this implicitly learned knowledge can be utilized quite efficiently in standard reasoning problems. The

(possibly exponential-size) KB itself will appear only in our analysis of a combined system for learning and reasoning.

Technically, our contribution is that we exhibit computationally efficient algorithms for reasoning under PAC-Semantics using both explicitly given rules and rules that are learned implicitly from partially obscured examples. As is typical for such works, we can illustrate the task performed by our algorithm with a story about an aviary. Suppose that we know that the birds of the aviary fly unless they are penguins, and that penguins eat fish. Now, suppose that we obtain a feeding log, from which we can glean that most (but perhaps not all) of the birds in the aviary seem not to eat fish. From this information, we can infer that most of the birds in the aviary can fly. Our algorithms, given the prior knowledge and partial information of the feeding log as input, will draw such a conclusion. In particular, they achieve this *without* being told up front that “not eating fish” is the key property that must be learned from the data; in an implicit sense, this premise is automatically picked out from among the myriad possible facts and rules that could be learned.

The conclusion that the birds of the aviary fly draws on both the empirical (partial) information and reasoning from our explicit, factual knowledge: on the one hand, our feeding log did not mention anything about whether or not the birds of the aviary could fly, and on the other hand, although our knowledge is sufficient to conclude that the birds that don’t eat fish can fly, it isn’t sufficient to conclude whether or not, broadly speaking, the birds in the aviary can fly. The key definition of “*witnessed evaluation*” of the (common sense) facts and rules is what makes this connection possible. It allows us to guarantee that we learn facts such as that the birds don’t eat fish when they can almost always be verified under the partial information, and at the same time, it enables the learned facts and rules to be easily integrated into derivations in standard proof systems. The underlying definition is standard in proof complexity, but this is the first work (to our knowledge) to use it to guarantee learnability in a partial information context.

Our results show that essentially all natural tractable proof systems considered in the literature (e.g., width-bounded and treelike resolution) can also utilize knowledge that is implicitly learned from partial examples to produce PAC conclusions, without losing the tractability of the systems. The penalty for access to such implicit knowledge is essentially independent of the size of the implicit KB. We also note that

*Supported by ONR grant number N000141210358 and NSF grant number CCF-0939370. This work was done while the author was also affiliated with MIT.

the introduction of probability to cope with the imperfection of learned rules does not harm the tractability of inference, in contrast to some richer probability logics [Halpern, 1990].

It is perhaps more remarkable in from a learning theoretic perspective that our approach does not require the rules to be learned (or discovered) to be completely consistent with the examples drawn from the (arbitrary) distribution. In the usual learning context, this would be referred to as *agnostic learning*, as introduced by Kearns et al. [1994]. It is essential to common sense reasoning that we can detect and utilize such rules that may even possess some known counterexamples, such as “birds fly.” But, agnostic learning is notoriously hard—Kearns et al. noted that even agnostic learning of clauses (over an arbitrary distribution, in the standard PAC-learning sense) would yield an efficient algorithm for PAC-learning DNF (also over arbitrary distributions), which remains the central open problem of computational learning theory. Again, by declining to produce a hypothesis, we manage to circumvent a key barrier (to the state of the art, at least).

1.1 Relationship to other work

Given that the task we consider is fundamental and has a variety of applications, other frameworks that can handle our task have naturally been proposed—for example, Markov Logic [Richardson and Domingos, 2006] is one well-known framework based on graphical models, and Bayesian Logic Programming [Kersting and De Raedt, 2008] is an approach that has grown out of the Inductive Logic Programming (ILP) community that can be used to achieve the tasks we consider here. The main distinction between all of these approaches and our approach is that these other approaches *all aim to model the distribution of the data*, which is generally a *much* more demanding task – both in terms of the amount of data and computation time required – than simply answering a query. Naturally, the upshot of these other works is that they are much richer and more versatile, and there are a variety of other tasks (e.g., density estimation, maximum likelihood computations) and/or settings (e.g., infinite first-order domains) that these frameworks can handle that we do not. Our aim is instead to show how this more limited (but still useful) task can be done much more efficiently, much like how algorithms such as SVMs and boosting can succeed at predicting attributes for concept classes of limited scope without needing to model the distribution of the data.

In this respect, our work is similar to the Learning to Reason framework of Khardon and Roth [1997], who showed how an NP-hard reasoning task (deciding a $\log n$ -CNF query), when coupled with a learning task beyond the reach of the state of the art (learning DNF from random examples) could result in an efficient overall system. The distinction between our work and Khardon and Roth’s is, broadly speaking, that we re-introduce the theorem-proving aspect that Khardon and Roth had explicitly sought to avoid. Briefly, these techniques permit us to incorporate declaratively specified background knowledge and moreover, permit us to cope with partial information in more general cases than Khardon and Roth [1999], who could only handle constant width clauses. As we noted earlier, we believe that reasoning is primarily of interest in partial information settings (cf. it is somewhat

subtle to explain what the technical contribution of Khardon and Roth [1997] achieves that is not achieved by a simple estimator). Another difference between our work and that of Khardon and Roth, that also distinguishes our work from traditional ILP (e.g., [Muggleton and De Raedt, 1994]), is that as mentioned above, we are able to utilize rules that hold with less than perfect probability (akin to agnostic learning, but easier to achieve here).

2 Definitions and preliminaries

2.1 PAC-Semantics

Inductive generalization (as opposed to deduction) inherently entails the possibility of making mistakes. Thus, the kind of rules produced by learning algorithms cannot hope to be valid in the traditional (Tarskian) sense (for reasons we describe momentarily), but intuitively they do capture some useful quality. PAC-Semantics were introduced by Valiant [2000] to capture in general this quality possessed by the output of PAC-learning algorithms when formulated in a logic. Precisely, suppose that we observe examples independently drawn from a distribution over $\{0, 1\}^n$; now, suppose that our algorithm has found a rule $f(x)$ for predicting some target attribute x_t from the other attributes. The formula “ $x_t = f(x)$ ” may not be valid in the traditional sense, as PAC-learning does not guarantee that the rule holds for every possible binding, only that the rule f so produced agrees with x_t with probability $1 - \epsilon$ with respect to future examples drawn from the same distribution. That is, the formula is instead “valid” in the following sense:

Definition 1 ($(1 - \epsilon)$ -valid) *Given a distribution D over $\{0, 1\}^n$, we say that a relation R is $(1 - \epsilon)$ -valid if $\Pr_{x \in D}[R(x) = 1] \geq 1 - \epsilon$.*

Of course, we may consider $(1 - \epsilon)$ -validity of relations R that are not obtained by learning algorithms and in particular, not of the form “ $x_t = f(x)$.”

A number of well-known specific examples of common sense reasoning were demonstrated to be captured by learned rules by Valiant [1995] and Roth [1995], and Valiant [2006] proposed learned knowledge in general to be suitable for capturing certain kinds of common sense knowledge. The non-monotonic examples discussed by Valiant and Roth arise when we consider conditioning D on some relatively rare event, such as encountering a penguin. Tolerance to counterexamples from such rare but possible events is thus crucial for capturing common sense reasoning. We won’t further consider how or when to condition on such events here. Instead, we take some such distribution as given and attempt to reason about typical examples from the distribution.

Valiant [2000] considered one rule of inference, *chaining*, for formulas of the form $\ell_t = f(x)$ where f is a linear threshold function: given a collection of literals such that the partial assignment obtained from satisfying those literals guarantees f evaluates to true, infer the literal ℓ_t . Valiant observed that for such learned formulas, the conjunction of literals derived from a sequence of applications of chaining is also $(1 - \epsilon')$ -valid for some polynomially larger ϵ' . It turns out that this property of soundness under PAC-Semantics is not a special feature of chaining: generally, it follows from the union

bound that any classically sound derivation is also sound under PAC-Semantics in a similar sense.

Proposition 2 (Classical reasoning in PAC-Semantics)

Let ψ_1, \dots, ψ_k be formulas such that each ψ_i is $(1 - \epsilon_i)$ -valid under a common distribution D for some $\epsilon_i \in [0, 1]$. Suppose that $\{\psi_1, \dots, \psi_k\} \models \varphi$ (in the classical sense). Then φ is $(1 - \epsilon')$ -valid under D for $\epsilon' = \sum_i \epsilon_i$.

So, soundness under PAC-Semantics does not pose any constraints on the rules of inference that we might consider; the degree of validity of the conclusions merely aggregates any imperfections in the various individual premises involved. We also note that without further knowledge of D , the loss of validity from the use of a union bound is optimal.

Subsequently, we will assume that our Boolean functions will be given by formulas of propositional logic formed over Boolean variables $\{x_1, \dots, x_n\}$ by negation and the following linear threshold connectives (which we will refer to as the *threshold basis* for propositional formulas):

Definition 3 (Threshold connective) A threshold connective for a list of k formulas ϕ_1, \dots, ϕ_k is given by a list of $k+1$ rational numbers, c_1, \dots, c_k, b . The formula $[\sum_{i=1}^k c_i \phi_i \geq b]$ is interpreted as follows: given a Boolean interpretation for the k formulas, the connective is true if $\sum_{i:\phi_i=1} c_i \geq b$.

Naturally, a threshold connective expresses a k -ary AND connective by taking the $c_i = 1$, and $b = k$, and expresses a k -ary OR by taking $c_1, \dots, c_k, b = 1$. By using the threshold connective as our basis, we will also be able to easily demonstrate that our results apply to some numerical proof systems such as cutting planes [Cook *et al.*, 1987], which reasons about systems of linear inequalities, and the polynomial calculus [Clegg *et al.*, 1996] a proof system for reasoning about the common roots of systems of polynomial equations.

We note that Valiant actually defines PAC-Semantics for first-order logic by considering D to be a distribution over the values of atomic formulas. He focuses on formulas of bounded arity over a polynomial size domain; then evaluating such formulas from the (polynomial size) list of values of all atomic formulas is tractable, and in such a case everything we consider here about propositional logic essentially carries over in the usual way, by considering each atomic formula to be a propositional variable (and rewriting the quantifiers as disjunctions or conjunctions over all bindings). As we don't have any insights particular to first-order logic to offer, we will focus exclusively on the propositional case in this work.

Michael [2008] built systems based on PAC-Semantics for addressing NLP tasks, providing some concrete demonstrations of how to apply the theory. More specifically, Michael and Valiant [2008] demonstrated that chaining with rules obtained from PAC-learning algorithms improved the accuracy of algorithms for predicting missing words while Michael [2009] described an approach to textual entailment.

2.2 Partial observability

Our knowledge of a domain will be obtained from a collection of *examples* independently drawn from a distribution D capturing our domain, and our main question of interest will be deciding whether or not a formula is $(1 - \epsilon)$ -valid in D .

But notice, in the story about the aviary in the introduction, reasoning was only needed on account of the *partial information* provided by the feeding log: if it had stated whether or not the birds could fly as well, then we could have answered this query more simply by a direct examination of the data. Generally, answering queries in PAC-Semantics from *complete* examples is trivial: Hoeffding's inequality guarantees that with high probability, the proportion of times that the query formula evaluates to 'true' is a good estimate of the degree of validity of the formula. Recall:

Theorem 4 (Hoeffding's inequality) Let X_1, \dots, X_m be i.i.d. random variables taking values in $[0, 1]$. Let $\bar{X} = \frac{1}{m} \sum_{i=1}^m X_i$. Then for every γ ,

$$\Pr[\bar{X} - \mathbb{E}[X_i] > \gamma] \leq e^{-2m\gamma^2}$$

In such cases, *reasoning per se* is of no use. We were interested in using common sense facts about our domain that were learnable rules over D in reasoning, so our focus necessarily concerns situations involving *both* learning and partial information, building on a theory developed by Michael [2010].

Definition 5 (Partial examples) A partial example ρ is an element of $\{0, 1, *\}^n$. We say that a partial example ρ is consistent with an example $x \in \{0, 1\}^n$ if whenever $\rho_i \neq *$, $\rho_i = x_i$.

Naturally, instead of examples directly from D , our knowledge of D will be derived from a collection of *partial examples* drawn from a *masking process* over D :

Definition 6 (Masking process) A mask is a function $m : \{0, 1\}^n \rightarrow \{0, 1, *\}^n$, with the property that for any $x \in \{0, 1\}^n$, $m(x)$ is consistent with x . A masking process M is a mask-valued random variable (i.e., a random function). We denote the distribution over partial examples obtained by applying a masking process M to a distribution D over assignments by $M(D)$.

Note that the definition of masking processes allows the hiding of entries to depend on the underlying example from D .

Of course, when e.g., *all* attributes of our examples are hidden by the masking process, any nontrivial knowledge of our domain is surely beyond "common sense." We therefore start by only considering some facts that can be easily empirically verified, effectively by generalizing Valiant's [2000] partial information chaining rule. (Simple consequences of such empirically grounded facts will also be included later.)

Definition 7 (Witnessed formulas) We define a formula to be witnessed to evaluate to true or false in a partial example by induction on its construction; we say that the formula is witnessed iff it is witnessed to evaluate to either true or false.

- A variable is witnessed to be true or false iff it is respectively true or false in the partial example.
- $\neg\phi$ is witnessed to evaluate to true iff ϕ is witnessed to evaluate to false; naturally, $\neg\phi$ is witnessed to evaluate to false iff ϕ is witnessed to evaluate to true.
- A formula with a threshold connective $[c_1\phi_1 + \dots + c_k\phi_k \geq b]$ is witnessed to evaluate to true iff

$$\sum_{i:\phi_i \text{ witnessed true}} c_i + \sum_{i:\phi_i \text{ not witnessed}} \min\{0, c_i\} \geq b$$

and it is witnessed to evaluate to false iff

$$\sum_{i:\phi_i \text{ witnessed true}} c_i + \sum_{i:\phi_i \text{ not witnessed}} \max\{0, c_i\} < b.$$

(i.e., iff the truth or falsehood, respectively, of the inequality is determined by the witnessed formulas, regardless of what values are substituted for the non-witnessed formulas.)

The formulas that are witnessed true with probability $1 - \epsilon$ will be our easily verified rules: the recursive definition gives a linear-time algorithm for computing the witnessed evaluation. They will thus also be the common sense facts that we will guarantee to be learnable. We stress that such formulas may still be (witnessed, even) false with probability up to ϵ . This tolerance to a few counterexamples is essential for them to capture interesting examples of common sense facts.

A formal example of particular interest is a CNF formula. A CNF is witnessed to evaluate to true in a partial example precisely when every clause has some literal that is satisfied. It is witnessed to evaluate to false precisely when there is some clause in which every literal is falsified. If we think of the clauses as encoding implications (e.g., Horn rules), then they are witnessed true when either the head is true and unmasked, or else when one of the body literals is false and unmasked; the conjunction of these rules is witnessed true when all of the individual rules are simultaneously witnessed true. Witnessed satisfaction is also necessary for a useful CNF (for resolution) to be guaranteed to be satisfied: When a clause is not witnessed true in a partial example and does not contain a complementary pair, in the absence of background knowledge, there is always a consistent example that falsifies it. By contrast, it is of course NP-complete to determine whether a CNF can be satisfied when it is not witnessed false. The definition then only “gives up” cases where no single clause is set to false, but nevertheless no consistent assignment across all of the clauses exists. This concession is significant in some richer proof systems that we consider later, such as k -DNF resolution [Krajíček, 2001].

Refining the motivating discussion somewhat, a witnessed formula is one that can be evaluated in a very straightforward, local manner. When the formula is *not* witnessed, we will likewise be interested in the following “simplification” of the formula obtained from an incomplete local evaluation:

Definition 8 (Restricted formula) Given a partial example ρ and a formula ϕ , the restriction of ϕ under ρ , denoted $\phi|_\rho$, is a formula recursively defined as follows:

- If ϕ is witnessed in ρ , then $\phi|_\rho$ is \top (“true”) if ϕ is witnessed true, and \perp (“false”) otherwise.
- If ϕ is a variable not set by ρ , $\phi|_\rho = \phi$.
- If $\phi = \neg\psi$ and ϕ is not witnessed in ρ , then $\phi|_\rho = \neg(\psi|_\rho)$.
- If $\phi = [\sum_{i=1}^k c_i \psi_i \geq b]$ and ϕ is not witnessed in ρ , suppose that ψ_1, \dots, ψ_ℓ are witnessed in ρ (and $\psi_{\ell+1}, \dots, \psi_k$ are not witnessed). Then $\phi|_\rho$ is $[\sum_{i=\ell+1}^k c_i (\psi_i|_\rho) \geq d]$ where $d = b - \sum_{i:\psi_i|_\rho = \top} c_i$.

For a restriction ρ and set of formulas F , we let $F|_\rho$ denote the set $\{\phi|_\rho : \phi \in F\}$.

Given unit cost arithmetic, restrictions are also easily computed from their definition in linear time. In a CNF that is not witnessed, the restriction simply deletes clauses that are witnessed satisfied in the partial example, and deletes the falsified literals from the remaining clauses.

2.3 Proof systems

The reasoning problems that we consider will be captured by “proof systems.” Formally:

Definition 9 (Proof system) A proof system is given by a sequence of relations $\{R_i\}_{i=0}^\infty$ over formulas such that R_i is of arity- $(i + 1)$ and whenever $R_i(\psi_{j_1}, \dots, \psi_{j_i}, \varphi)$ holds, $\{\psi_{j_1}, \dots, \psi_{j_i}\} \models \varphi$. Any formula φ satisfying R_0 is said to be an axiom of the proof system. A proof of a formula ϕ from a set of hypotheses H in the proof system is given by a finite sequence of triples consisting of

1. A formula ψ_k
2. A relation R_i of the proof system or the set H
3. A subsequence of formulas $\psi_{j_1}, \dots, \psi_{j_i}$ with $j_\ell < k$ for $\ell = 1, \dots, i$ (i.e., from the first components of earlier triples in the sequence) such that $R_i(\psi_{j_1}, \dots, \psi_{j_i}, \psi_k)$ holds, unless $\psi_k \in H$.

for which ϕ is the first component of the final triple in the sequence.

Needless to say it is generally expected that R_i is somehow efficiently computable, so that the proofs can be checked. We don’t explicitly impose such a constraint on the formal object for the sake of simplicity, but the reader should be aware that these expectations will be fulfilled in all cases of interest.

We will be interested in the effect of the restriction (partial evaluation) mapping applied to *proofs*—that is, the “projection” of a proof in the original logic down to a proof over the smaller set of variables by the application of the restriction to every step in the proof. Although it may be shown that this at least preserves the (classical) semantic soundness of the steps, this falls short of what we require: we need to know that the *rules of inference* are preserved under restrictions. Since the relations defining the proof system are arbitrary, though, this property must be explicitly verified. Formally, then:

Definition 10 (Restriction-closed proof system) We will say that a proof system over propositional formulas is restriction closed if for every proof of the proof system and every partial example ρ , for any (satisfactory) step of the proof $R_k(\psi_1, \dots, \psi_k, \phi)$, there is some $j \leq k$ such that for the subsequence $\psi_{i_1}, \dots, \psi_{i_j}$ $R_j(\psi_{i_1}|_\rho, \dots, \psi_{i_j}|_\rho, \phi|_\rho)$ is satisfied, and the formula \top (“true”) is an axiom.¹

So, when a proof system is restriction-closed, given a derivation of a formula φ from ψ_1, \dots, ψ_k , we can extract a derivation of $\varphi|_\rho$ from $\psi_1|_\rho, \dots, \psi_k|_\rho$ for any partial example ρ such that the steps of the proof consist of formulas mentioning only the variables masked in ρ . This means that we can extract a proof of a “special case” from a more general proof by applying the restriction operator to every formula

¹This last condition is a technical condition that usually requires a trivial modification of any proof system to accommodate. We can usually do without this condition in actuality, but the details depend on the proof system.

in the proof. An illustration of this transformation appears in Figure 1 in the next section, where we describe how the problem in the introduction is solved by our algorithm. It turns out to be folklore in the proof complexity community that almost every propositional proof system is restriction-closed—Beame et al. [2004] defined “natural” special cases of resolution to have this property, in particular.

We will be especially interested in special cases of the decision problem for a logic given by a collection of “simple” proofs—if the proofs are sufficiently restricted, it is possible to give efficient algorithms to search for such proofs, and then such a special case of the decision problem will be tractable, in contrast to the general case. Formally, now:

Definition 11 (Automatizability problem) *Fix a proof system, and let \mathcal{S} be a set of proofs in the proof system. The automatizability problem for \mathcal{S} is then the following promise problem: given as input a formula φ and a set of hypotheses H such that either there is a proof of φ in \mathcal{S} from H or else $H \not\models \varphi$, decide which case holds.*

A classic example of such an automatizability problem for which efficient algorithms exist is for formulas of propositional logic that have resolution derivations of constant width (first studied by Galil [1977]) using a simple dynamic programming algorithm. Essentially the same dynamic programming algorithm can be used to solve the automatizability problem for bounded-width k -DNF resolution, a slightly stronger proof system introduced by Krajíček [2001]. In a different strengthening, Clegg et al. [1996] showed that polynomial calculus has an efficient algorithm when the polynomials are all of degree bounded by some absolute constant d . Another example (of incomparable strength) is that treelike resolution proofs that can be derived while only remembering a constant number of clauses (i.e., of constant “clause space” [Esteban and Torán, 2001]) can also be found in polynomial time by a variant of Beame and Pitassi’s algorithm [1996] (noted by Kullman [1999]). We also note that the special case of the cutting planes proof system [Cook et al., 1987] in which the coefficient vectors have bounded L_1 -norm and constant sparsity is also suitable [Juba, 2012, Section 4.3]. Our approach is sufficiently general to show that implicit learning can be added to *all* of these special cases.

We will thus be interested in syntactic restrictions of restriction-closed proof systems like those above. We wish to know that (in contrast to the rules of the proof system) these *syntactic restrictions* are likewise closed under restrictions in the following sense:

Definition 12 (Restriction-closed set of proofs) *A set of proofs \mathcal{S} is said to be restriction closed if whenever there is a proof of a formula φ from a set of hypotheses H in \mathcal{S} , there is also a proof of $\varphi|_\rho$ from the set of hypotheses $H|_\rho$ in \mathcal{S} for any partial example ρ .*

It is not difficult to show that all of the syntactic special cases of the proof systems that feature efficient algorithms mentioned above are *also* restriction-closed sets of proofs (and as hinted at previously, this seems to be folklore). The full details for the systems we mentioned here are available in a technical report [Juba, 2012].

Algorithm 1: DecidePAC

parameter: Algorithm A solving the automatizability problem for the class of proofs \mathcal{S} .
input : Formula φ , $\epsilon, \delta, \gamma \in (0, 1)$, list of partial examples $\rho^{(1)}, \dots, \rho^{(m)}$ from $M(D)$, list of hypothesis formulas H
output : *Accept* if there is a proof of φ in \mathcal{S} from H and formulas ψ_1, ψ_2, \dots that are simultaneously witnessed true with probability at least $1 - \epsilon + \gamma$ on $M(D)$;
Reject if $H \Rightarrow \varphi$ is not $(1 - \epsilon - \gamma)$ -valid under D .

begin
 $FAILED \leftarrow 0$.
 foreach partial example $\rho^{(i)}$ in the list **do**
 if $A(\varphi|_{\rho^{(i)}}, H|_{\rho^{(i)}})$ rejects **then**
 Increment $FAILED$.
 if $FAILED > \lfloor \epsilon \cdot m \rfloor$ **then**
 return Reject
 return Accept

3 Inferences from incomplete data with implicit learning

In a given sample of data, there are potentially exponentially many possible different formulas that could be easily learned from the data. Specifically, this “common sense knowledge” consists of formulas which are easy to check for consistency with the underlying distribution by testing them on a sample of partial examples. More formally, these are formulas that are witnessed to evaluate to true on the distribution over partial examples with probability at least $(1 - \epsilon)$. We will see that it is possible to simulate access to *all* of these “common sense facts” during reasoning; whenever some collection of such “common sense facts” suffice to complete a proof of our query, our algorithm will accept the query. Notice that this may even include “facts” that may be witnessed false with probability up to ϵ . This is crucial if we are to draw an inference such as “birds fly” when our examples have a small but non-negligible chance of being penguins (and thus for non-monotonic effects to appear given such rare events).

We now state and prove the main theorem. It shows that a variant of the automatizability problem, in which the algorithm is given partial examples and expected to accept queries based on proofs that invoke such “common sense knowledge,” is essentially no harder than the original automatizability problem as long as the proof system is restriction-closed. The reduction is *very* simple and is given in Algorithm 1.

Theorem 13 (Implicit learning preserves tractability) *Let \mathcal{S} be a restriction-closed set of proofs for a restriction-closed proof system. Suppose that there is an algorithm for the automatizability problem for \mathcal{S} running in time $T(n, |\varphi|, |H|)$ on input φ and H over n variables. Let D be a distribution over examples, M be any masking process, and H be any set of formulas. Then there is an algorithm that, on input φ , H , δ and ϵ , uses $O(1/\gamma^2 \log 1/\delta)$ examples, runs in time*

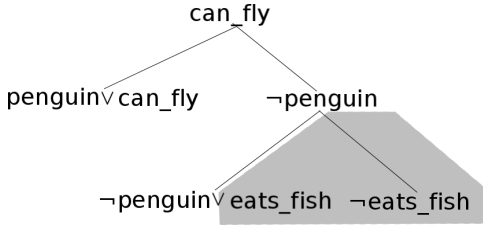


Figure 1: A resolution derivation of the example from the introduction. The shaded portion is omitted from the restriction of the proofs on the input examples (i.e., the log entries).

$O(T(n, |\varphi|, |H|) \frac{1}{\gamma^2} \log \frac{1}{\delta})$, and such that given that either

- $[H \Rightarrow \varphi]$ is not $(1 - \epsilon - \gamma)$ -valid with respect to D or
- there exists a proof of φ from $\{\psi_1, \dots, \psi_k\} \cup H$ in \mathcal{S} such that ψ_1, \dots, ψ_k are simultaneously witnessed to evaluate to true with probability $1 - \epsilon + \gamma$ over $M(D)$

decides which case holds with probability $1 - \delta$.

Proof: Suppose we run Algorithm 1 on $m = \frac{1}{2\gamma^2} \ln \frac{1}{\delta}$ examples drawn from $M(D)$. Then, (noting that we need at most $\log m$ bits of precision for $\lfloor \epsilon \cdot m \rfloor$) the claimed running time bound and sample complexity is immediate.

As for correctness, first note that by the soundness of the proof system, whenever there is a proof of $\varphi|_{\rho^{(i)}}$ from $H|_{\rho^{(i)}}$, $\varphi|_{\rho^{(i)}}$ must evaluate to true in any interpretation of the remaining variables consistent with $H|_{\rho^{(i)}}$. Thus, if $H \Rightarrow \varphi$ is not $(1 - \epsilon - \gamma)$ -valid with respect to D , an interpretation sampled from D must satisfy H and falsify φ with probability at least $\epsilon + \gamma$; for any partial example ρ derived from this interpretation (i.e., sampled from $M(D)$), the original interpretation is still consistent, and therefore $H|_{\rho} \not\models \varphi|_{\rho}$ for this ρ . So in summary, we see that a ρ sampled from $M(D)$ produces a formula $\varphi|_{\rho}$ such that $H|_{\rho} \not\models \varphi|_{\rho}$ with probability at least $\epsilon + \gamma$, and so the algorithm A rejects with probability at least $\epsilon + \gamma$. It follows from Hoeffding’s inequality now that for m as specified above, at least ϵm of the runs of A reject (and hence the algorithm rejects) with probability at least $1 - \delta$.

So, suppose instead that there is a proof in \mathcal{S} of φ from H and some formulas ψ_1, \dots, ψ_k that are all witnessed to evaluate to true with probability at least $1 - \epsilon + \gamma$ over $M(D)$. Then, with probability $1 - \epsilon + \gamma$, $\psi_1|_{\rho}, \dots, \psi_k|_{\rho} = \top$. Then, since \mathcal{S} is a restriction closed set, if we replace each assertion of some ψ_j with an invocation of R_0 for the axiom \top , then by applying the restriction ρ to every formula in the proof, one can obtain a proof of $\varphi|_{\rho}$ from $H|_{\rho}$ alone. Therefore, as A solves the automatizability problem for \mathcal{S} , we see that for each ρ drawn from $M(D)$, $A(\varphi|_{\rho}, H|_{\rho})$ must accept with probability at least $(1 - \epsilon + \gamma)$, and Hoeffding’s inequality again gives that the probability that more than ϵm of the runs reject is at most δ for this choice of m . ■

An illustration. We now describe how Algorithm 1 solves the aviary problem described in the introduction. Suppose that in 99% of the log entries, the bird is observed to eat something other than fish; the literal expressing that a bird

in question doesn’t eat fish is then witnessed true in a 99% fraction of the examples. This fact, together with our prior knowledge that the penguins eat fish and that all of the birds that aren’t penguins can fly, is sufficient to complete a simple treelike resolution derivation of the conclusion that the bird in question can fly (the underlying proof appears in Figure 1). We now consider the partial examples obtained from the log entries: in 99% of these partial examples, the literal expressing that the bird eats fish is (known to be) false. Under the restrictions corresponding to such log entries, one obtains a resolution derivation that only invokes the restrictions of the background knowledge clauses as premises, as illustrated in Figure 1. If we run Algorithm 1 with a query corresponding to the literal “can fly,” given as input the log examples, the background knowledge clauses, and $\epsilon = .05$, then since a proof is found in more than 95% of the examples, the algorithm accepts, concluding that (most of) the birds can fly. Theorem 13 in turn guarantees that the statistical conclusions drawn by the algorithm accurately reflect the underlying composition of the aviary if there are sufficiently many examples.

The necessity of computationally feasible witnessing. Although our choice of witnessed formulas as the “common sense knowledge” that should be learned may seem like an ad hoc choice, we note that it is merely a “base case” for the learning—that is, it follows from our requirements on the algorithms in each case that any other knowledge that follows from a simple derivation over these witnessed formulas must also be discoverable by the algorithm. Thus, for each choice of proof system, there are (potentially) different collections of formulas that the algorithm is expected to be capable of discovering. We note that in any case, we require *some* collection of formulas for which $(1 - \epsilon)$ -validity is computationally feasible to verify: the specification of the problem requires that if we invoke our (efficient) algorithm on a query corresponding to some “common sense fact,” the algorithm must verify that it is $(1 - \epsilon)$ -valid (which may be strengthened to verification on individual partial examples [Juba, 2012, Appendix B]). Witnessed evaluation is at least a natural class of formulas for which this test is possible (and easy), whereas many other natural classes, notably the class of formulas whose restrictions are tautologies as considered by Michael [2010], may be computationally infeasible to test, and thus inappropriate for our purposes.

4 Directions for future work

A possible direction for future work raised directly by this work involves the development of algorithms for reasoning in PAC-Semantics *directly*, that is, not obtained by applying Theorem 13 to algorithms for the automatizability problems under the classical (worst-case) semantics of the proof systems. We will elaborate on some possible starting points next.

4.1 Incorporating explicit learning

One approach concerns modern algorithms for deciding satisfiability; a well-known result due to Beame et al. [2004] establishes that these algorithms effectively perform a search

for resolution proofs of unsatisfiability (or, satisfying assignments), and work by Atserias et al. [2011] shows that these algorithms (when they make certain choices at random) are effective for deciding bounded-width resolution.

The overall architecture of these modern “SAT-solvers” largely follows that of Zhang et al. [2001], and is based on improvements to DPLL [Davis and Putnam, 1960; Davis et al., 1962] explored earlier in several other works [Marques-Silva and Sakallah, 1999; Bayardo Jr. and Schrag, 1997; Gomes et al., 1997]. Roughly speaking, the algorithm makes an arbitrary assignment to an unassigned variable, and then examines what other variables must be set in order to satisfy the formula; when a contradiction is entailed by the algorithm’s decision, a new clause is added to the formula (entailed by the existing clauses) and the search continues on a different setting of the variables. A few simple rules are used for the task of exploring the consequences of a partial setting of the variables—notably, for example, *unit propagation*: whenever all of the literals in a clause are set to false except for one (unset) variable, that final remaining literal must be set to true if the assignment is to satisfy the formula.

One possibility for improving the power of such algorithms for reasoning under PAC-Semantics using examples is that one might wish to use an explicit learning algorithm such as WINNOW [Littlestone, 1988] to learn additional (approximately valid) rules for extending partial examples. If we are using these algorithms to find resolution refutations, then when a refutation was produced by such a modified architecture, it would establish that the input formula is only satisfied with some low probability (depending on the error of the learned rules that were actually invoked during the algorithm’s run).

Given such a modification, one must then ask: does it actually improve the power of such algorithms? Work by Pipatsrisawat and Darwiche [2011] (related to the above work) has shown that with appropriate (nondeterministic) guidance in the algorithm’s decisions, such algorithms do actually find general (i.e., DAG-like) resolution proofs in a polynomial number of iterations. Yet, it is still not known whether or not a feasible decision strategy can match this. Nevertheless, their work (together with the work of Atserias et al.) provides a potential starting point for such an analysis.

A suggestion for empirical work

Another obvious direction for future work is the development and tuning of real systems for inference in PAC-Semantics. While the algorithms we have presented here illustrate that such inference can be theoretically rather efficient and are evocative of how one might approach the design of a real-world algorithm, the fact is that (1) any off-the-shelf SAT solver can be easily modified to serve this purpose and (2) SAT solvers have been highly optimized by years of effort. It would be far easier and more sensible for a group with an existing SAT solver implementation to simply make the following modification, and see what the results are: along the lines of Algorithm 1, for a sample of partial examples $\{\rho^{(1)}, \dots, \rho^{(m)}\}$, the algorithm loops over $i = 1, \dots, m$, taking the unmasked variables in $\rho^{(i)}$ as decisions and checks for satisfiability with respect to the remaining variables. Count-

ing the fraction of the partial examples that can be extended to satisfying assignments then gives a bound on the validity of the input formula. Crucially, in this approach, *learned clauses are shared across examples*. Given that there is a common resolution proof across instances (cf. the connection between SAT solvers and resolution [Beame et al., 2004]) we would expect this sharing to lead to a faster running time than simply running the SAT solver as a black box on the formulas obtained by “plugging in” the partial examples (although that is another approach).

4.2 Exploiting limited kinds of masking processes

Another direction for possibly making more sophisticated use of the examples in reasoning under PAC-Semantics involves restricting the masking processes. In the pursuit of reasoning algorithms, it might be helpful to consider restrictions that allow some possibility of “extrapolating” from the values of variables seen on one example to the values of hidden variables in other examples, which is not possible in general since the masking process is allowed to “see” the example before choosing which entries to mask. (Relatedly, work by Bonet et al. [2004] shows, under cryptographic assumptions, that when some attributes are always masked, the automatizability problem for systems such as bounded-depth Frege can’t be solved efficiently for some distributions.) If the masks were chosen independently of the underlying examples and occasionally revealed every attribute, this might enable such guessing to be useful. Some preliminary results in this direction have been obtained when the learning problem is restricted (to learning a system of parity constraints over a uniform distribution) and the masking process decides whether to mask each attribute by tossing a biased coin: the automatizability problem for general resolution can then be decided in quasipolynomial time [Juba, 2013].

4.3 Query-driven explicit learning

A final question is whether or not it might be possible to extend Algorithm 1 to produce an explicit proof from an explicit set of formulas that are satisfied with high probability from e.g., algorithms for finding treelike resolution proofs even when the CNF we need is not 1-valid. (It is generally easy to find $(1 - \epsilon)$ -valid premises when 1-valid premises exist by simply testing for consistency with the partial examples, given *bounded concealment* in the sense of Michael [2010].) Although this is a somewhat ambitious goal, if one takes Algorithm 1 as a starting point, the problem is of a similar form to one considered by Dvir et al. [2012]—there, they considered learning decision trees from restrictions of the target tree. The main catch here is that in contrast to their setting, we are not guaranteed that we find restrictions of the same underlying proof, even when one is assumed to exist.

Acknowledgements

This work was heavily influenced by conversations with Leslie Valiant. I thank the anonymous reviewers for their constructive comments.

References

- [Atserias *et al.*, 2011] Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. *JAIR*, 40:353–373, 2011.
- [Bayardo Jr. and Schrag, 1997] Roberto J. Bayardo Jr. and Robert C. Schrag. Using CSP look-back techniques to solve real-world SAT instances. In *Proc. 14th AAAI*, pages 203–208, 1997.
- [Beame and Pitassi, 1996] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *Proc. 37th FOCS*, pages 274–282, 1996.
- [Beame *et al.*, 2004] Paul Beame, Henry Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *JAIR*, 22:319–351, 2004.
- [Bonet *et al.*, 2004] Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldá, Alexis Maciel, and Toniann Pitassi. Non-automatizability of bounded-depth Frege proofs. *Comput. Complex.*, 13:47–68, 2004.
- [Clegg *et al.*, 1996] Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the Gröbner basis algorithm to find proofs of unsatisfiability. In *Proc. 28th STOC*, pages 174–183, 1996.
- [Cook *et al.*, 1987] W. Cook, C. R. Coullard, and G. Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987.
- [Davis and Putnam, 1960] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *JACM*, 7(3):201–215, 1960.
- [Davis *et al.*, 1962] Martin Davis, George Logemann, and Donald W. Loveland. A machine program for theorem-proving. *CACM*, 5(7):394–397, 1962.
- [Dvir *et al.*, 2012] Zeev Dvir, Anup Rao, Avi Wigderson, and Amir Yehudayoff. Restriction access. In *Proc. 3rd ITCS*, 2012.
- [Esteban and Torán, 2001] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Inf. Comp.*, 171(1):84–97, 2001.
- [Galil, 1977] Zvi Galil. On resolution with clauses of bounded size. *SIAM J. Comput.*, 6:444–459, 1977.
- [Gomes *et al.*, 1997] Carla P. Gomes, Bart Selman, and Nuno Crato. Heavy-tailed distributions in combinatorial search. In *Proc. 3rd Int’l Conf. on Principles and Practice of Constraint Programming (CP97)*, volume 1330 of *LNCS*, pages 121–135. Springer, 1997.
- [Halpern, 1990] Joseph Y. Halpern. An analysis of first-order logics of probability. *Artif. Intel.*, 46:311–350, 1990.
- [Juba, 2012] Brendan Juba. Learning implicitly in reasoning in PAC-semantics. arXiv:1209.0056v1 [cs.AI], 2012.
- [Juba, 2013] Brendan Juba. PAC quasi-automatizability of resolution over restricted distributions. arXiv:1304.4633 [cs.DS], 2013.
- [Kearns *et al.*, 1994] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Towards efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.
- [Kersting and De Raedt, 2008] Kristian Kersting and Luc De Raedt. Basic principles of learning bayesian logic programs. In Luc De Raedt, Paolo Frasconi, Kristian Kersting, and Stephen Muggleton, editors, *Probabilistic Inductive Logic Programming: Theory and Applications*, volume 4911 of *LNCS*, pages 189–221. Springer, 2008.
- [Khardon and Roth, 1997] Roni Khardon and Dan Roth. Learning to reason. *J. ACM*, 44(5):697–725, 1997.
- [Khardon and Roth, 1999] Roni Khardon and Dan Roth. Learning to reason with a restricted view. *Machine Learning*, 35:95–116, 1999.
- [Krajíček, 2001] Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170:123–140, 2001.
- [Kullmann, 1999] Oliver Kullmann. Investigating a general hierarchy of polynomially decidable classes of CNF’s based on short tree-like resolution proofs. Technical Report TR99-041, ECCC, 1999.
- [Lenat, 1995] Douglas B. Lenat. CYC: a large-scale investment in knowledge infrastructure. *CACM*, 38(11):33–38, 1995.
- [Littlestone, 1988] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learn.*, 2(4):285–318, 1988.
- [Marques-Silva and Sakallah, 1999] João P. Marques-Silva and Karem A. Sakallah. GRASP: a search algorithm for propositional satisfiability. *IEEE Trans. Comput.*, 48(5):506–521, 1999.
- [McCarthy, 1959] John McCarthy. Programs with common sense. In *Teddington Conf. on the Mechanization of Thought Processes*, pages 756–791, 1959. Available at <http://www-formal.stanford.edu/jmc/mcc59.html>.
- [Michael and Valiant, 2008] Loizos Michael and Leslie G. Valiant. A first experimental demonstration of massive knowledge infusion. In *Proc. 11th KR*, pages 378–389, 2008.
- [Michael, 2008] Loizos Michael. *Autodidactic Learning and Reasoning*. PhD thesis, Harvard University, 2008.
- [Michael, 2009] Loizos Michael. Reading between the lines. In *Proc. 21st IJCAI*, pages 1525–1530, 2009.
- [Michael, 2010] Loizos Michael. Partial observability and learnability. *Artif. Intel.*, 174(11):639–669, 2010.
- [Muggleton and De Raedt, 1994] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *J. Logic Programming*, 19:629–679, 1994.
- [Pipatsrisawat and Darwiche, 2011] Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artif. Intel.*, 175:512–525, 2011.
- [Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Mach. Learn.*, 62:107–136, 2006.
- [Roth, 1995] Dan Roth. Learning to reason: the non-monotonic case. In *Proc. 14th IJCAI*, volume 2, pages 1178–1184, 1995.
- [Spelke and Kinzler, 2007] Elizabeth S. Spelke and Katherine D. Kinzler. Core knowledge. *Developmental Science*, 10(1):89–96, 2007.
- [Stork, 1999] David G. Stork. The open mind initiative. *IEEE Expert Systems and Their Applications*, 14(3):16–20, 1999.
- [Valiant, 1995] Leslie G. Valiant. Rationality. In *Proc. 8th COLT*, pages 3–14, 1995.
- [Valiant, 2000] Leslie G. Valiant. Robust logics. *Artif. Intel.*, 117:231–253, 2000.
- [Valiant, 2006] Leslie G. Valiant. Knowledge infusion. In *Proc. AAAI’06*, pages 1546–1551, 2006.
- [Zhang *et al.*, 2001] Lintao Zhang, Conor F. Madigan, Matthew W. Moskewicz, and Sharad Malik. Efficient conflict driven learning in a Boolean satisfiability solver. In *Proc. IEEE/ACM Int’l Conf. on Computer Aided Design (ICCAD’01)*, pages 279–285, 2001.