Forming Concepts for Fast Inference

Henry Kautz and Bart Selman
AI Principles Research Department
AT&T Bell Laboratories
600 Mountain Avenue, Room 2C-407
Murray Hill, NJ 07974
{kautz, selman}@research.att.com

Abstract

Knowledge compilation speeds inference by creating tractable approximations of a knowledge base, but this advantage is lost if the approximations are too large. We show how learning concept generalizations can allow for a more compact representation of the tractable theory. We also give a general induction rule for generating such concept generalizations. Finally, we prove that unless NP $\subseteq$ non-uniform P, not all theories have small Horn least upper-bound approximations.

1 Introduction

Work in machine learning has traditionally been divided into two main camps: concept learning (e.g. [Kearns, 1990]) and speed-up learning (e.g. [Minton, 1988]). The work reported in this paper bridges these two areas by showing how concept learning can be used to speed up inference by allowing a more compact and efficient representation of a knowledge base.

We have been studying techniques for boosting the performance of knowledge representation systems by compiling expressive but intractable representations into a computationally efficient form. Because the output representation language is, in general, strictly less expressive than the input (source) language, the output is an approximation of the input, rather than an exact translation. We call this process *knowledge compilation by theory approximation.*

For example, it is NP-hard to determine if a clausal query follows from a theory represented by propositional clauses, and thus all foreseeable algorithms for this problem require time exponential in the size of the theory in the worst case. On the other hand, the problem can be solved in linear time for theories expressed by Horn clauses. We have developed algorithms for computing two kinds of Horn approximations to general clausal theories [Selman and Kautz, 1991; Kautz and Selman, 1991]. The first is a *Horn lower-bound,* defined as a set of Horn clauses that entails the source theory. We proved that a best (logically weakest) such bound, called a *Horn greatest lower-bound* (GLB), can be always be represented by a set of Horn clauses no larger than the source theory. Thus such a GLB can be used to quickly determine if a query does *not* follow from the source theory: If the query does not follow from the GLB (which can be checked in linear time), it does not follow from the source theory.

This paper appears in the *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI–92)*, San Jose, CA, July 1992.

The second kind of approximation is a Horn upper-bound, defined as a set of Horn clauses that is entailed by the source theory. The best (logically strongest) such set is called the *Horn least upper-bound* (LUB).[1] The LUB can be used to quickly determine if a query *does* follow from the source theory: If the query *does* follow from the LUB (which can be checked in linear time), it also follows from the source theory.

If the query does follow from the GLB but does not follow from the LUB, then the bounds cannot be used to answer it. The system can either return "unknown" or choose to do full theorem proving with the original theory. In either case, the resulting system is sound: use of the bounds speeds inference by allowing some proportion of the queries to be answered quickly, but introduces no erroneous answers. It may take a great deal of time to actually compute the bounds, but this cost is "off-line": our goal is to have fast "run-time" query answering. In the papers cited above we also show how the bounds can be computed and used in an incremental, anytime fashion.

Our original papers on this work left open the question of the worst-case size of the LUB relative to that of the source theory. Although it is easy to show that the LUB is sometimes equivalent to a set of Horn clauses which is exponential in the size of the source theory, that alone does not rule out the possibility that there always exists an alternative Horn axiomatization of that same LUB which is not significantly larger than the source theory. In this paper we exhibit a clausal theory that shows such an exponential blowup can be inherent: that is, the smallest Horn theory that is equivalent to the LUB is exponentially larger than the source theory.

This would appear to be very bad news if one hoped to use the LUB as a way to speed up inference, since the increase in size can offset the decrease in inference time. But in fact this negative result leads to an interesting insight about theory approximation: minor changes to the source theory can dramatically reduce the size of the minimum representation of LUB. In particular, adding simple concept definitions to a source theory can sometimes decrease the size of the LUB by an exponential factor. The definitions add no new information about the world; the original and augmented source theories agree on all formulas that do not contain the newly-defined concepts. Furthermore, we can present an intuitively plausible rule for automatically generating such definitions.

We cannot prove that the techniques described in this paper for creating a compact representation of the LUB are complete; indeed, we believe that are simply useful heuristics. In fact, we will prove that unless a very surprising result holds in the complexity of finite functions, compact and tractable representations of the LUB do not always exist.

The connection between concept learning and speed-up learning has always been implicit in much of the work in the two fields. For example, work on algorithms for learning decision trees [Pagallo and Haussler, 1990] has the goal of generating trees that are small, and/or are expected to classify objects with a minimum number of tests. In work on speed-up learning for problem-solving, the learned macro-operators can also be considered to be newly defined concepts [Minton, 1985]. Our work arose in the context of developing tractable approximations of a knowledge base, and differs from first kind of example in that no new information is given to the system during the learning process, and differs from the second in that the concepts are induced before any problem instances (in our case, queries to the knowledge base) are presented to the system. Work in reformulation [Amarel, 1968; Bresina *et al.*, 1986; Subramanian and Genesereth, 1987] is somewhat similiar in that reformulating a problem to make it easier to solve may also involve the introduction of new terms. Most of the work in that area, however, tries to find efficient reformulations of particular problem instances, rather than complete theories,

[1] Throughout this paper we use LUB to mean *Horn* least upper-bound, although in the final section we briefly discuss some other kinds of least upper-bounds.

and does not try to trade off-line costs for run-time efficiency.

The paper is structured as follows. First we outline knowledge compilation using Horn approximations. The next section presents an example of a theory with an exponential LUB. Then we show that the same theory with an added defined concept has a small LUB (polynomial in the size of the source theory). The following section deals with issue of inducing the necessary concepts, and introduces a method of compacting a theory (i.e. the LUB) by inducing concepts. Finally we prove that unless the polynomial hierarchy collapses to Σ_2, there must always be some theories whose LUB cannot be represented in a form that is both small and tractable.

2 Theory Approximation

For a full introduction to knowledge compilation using Horn approximations see [Selman and Kautz, 1991]; a generalization to other kinds of approximations and the first-order case appears in [Kautz and Selman, 1991; Greiner, 1992]. This section summarizes only the relevant definitions and results.

We assume a standard propositional language, and use p, q, r, and s to denote propositional letters. A *literal* is either a propositional letter, called a positive literal, or its negation, called a negative literal, and is represented by w, x, y, or z. A *clause* is a disjunction of literals, and can be represented by the set of literals it contains. Clauses are sometimes written using material implication to make their intended meaning clear; for example, the clauses $p \vee \neg q \vee \neg r$ and $q \supset (p \vee \neg r)$ are identical. Greek letters α and β are used to represent clauses or parts of clauses (disjunctions of literals).

A set of clauses is called a *clausal theory*, and is represented by the Greek letters Σ or θ. A clause is *Horn* if and only if it contains at most one positive literal; a set of such clauses is called a *Horn theory*. (Note that we are not restricting our attention to definite clauses, which contain exactly one positive literal; a Horn clause may be completely negative.) A set of clauses Σ *entails* a set of clauses Σ', written $\Sigma \models \Sigma'$, if every model (satisfying truth-assignment) of Σ is a model of Σ'. The general problem of determining if a given clause is entailed by clausal theory is NP-hard [Cook, 1971], and thus almost certainly intractable. However, the problem for Horn theories can be solved in time linear in the combined lengths of the theory and query [Dowling and Gallier, 1984]. (Note that the query need not be Horn; in fact, the problem remains linear for even broader clauses of queries, such as arbitrary CNF formulas, and DNF formulas, where each disjunct contains at most one negative literal.)

Following are the definitions of the Horn upper-bound approximations of a clausal theory, as described in the introduction.

Definition: Horn Upper-Bound and LUB
Let Σ be a set of clauses. A set Σ_{ub} of Horn clauses is a Horn upper-bound of Σ iff $\Sigma \models \Sigma_{\text{ub}}$. A set Σ_{lub} of Horn clauses is a Horn least upper-bound (LUB) of Σ iff it is a Horn upper-bound, and there is no Horn upper-bound Σ_{ub} such that $\Sigma_{\text{ub}} \models \Sigma_{\text{lub}}$ and $\Sigma_{\text{lub}} \not\models \Sigma_{\text{ub}}$.

For example, let Σ be $\{p \vee q,\ (p \wedge r) \supset s,\ (q \wedge r) \supset s\}$. Then one Horn upper-bound is $\{(p \wedge r) \supset s,\ (q \wedge r) \supset s\}$, and the LUB is $\{r \supset s\}$.

The LUB of a theory is unique up to logical equivalence. That is, there may be distinct sets of Horn clauses Σ_{lub} and Σ'_{lub} that satisfy the conditions stated above, but if so, $\Sigma_{\text{lub}} \models \Sigma'_{\text{lub}}$ and $\Sigma'_{\text{lub}} \models \Sigma_{\text{lub}}$. It is important to note that such distinct representations of the LUB may vary greatly in size: for example, one may be the size of Σ, and the other of size $2^{|\Sigma|}$.

The LUB of a theory can be used as a quick but incomplete method of testing if the theory entails a query α by the following observations:

- If $\Sigma_{\text{lub}} \models \alpha$ then $\Sigma \models \alpha$.
- If α is Horn, then $\Sigma_{\text{lub}} \models \alpha$ if and only if $\Sigma \models \alpha$.

The LUB of a theory can be computed by resolution. The basic method is to generate all resolvants of Σ (a finite set, since the language is propositional), and then eliminate all non-Horn clauses. The result is a representation of the LUB, but it will usually contain many redundant clauses — for example, $p \supset q$ and $q \supset r$ as well as $p \supset r$. The representation can be minimized by repeatedly striking out any clause that is entailed by all the other clauses in the set. This basic algorithm can be optimized so that it generates fewer redundant clauses, but there is little hope for a polynomial time algorithm since the problem is NP-hard ([Selman and Kautz, 1991]). We accept this potential cost, however; the game we are playing is to see how fast we can make run-time question-answering by moving computational effort to a pre-processing stage. In addition, the knowledge compilation algorithms naturally generate a sequence of approximations that converge to the true LUB and GLB. These intermediate approximations can be used for question-answering even before the algorithm halts.

3 Explosion of the LUB

Knowledge compilation provides the greatest advantage when the representation of the LUB is as small as possible. Is it always possible to find a representation of the LUB which is of comparable size to that of the source theory? The answer is no, as the following example demonstrates.

The source theory contains the following clauses, which can be interpreted as rules for deciding if someone is a cognitive scientist. The clauses are numbered for reference.

$$(\text{CompSci} \wedge \text{Phil} \wedge \text{Psych}) \supset \text{CogSci} \tag{1}$$
$$\text{ReadsMcCarthy} \supset (\text{CompSci} \vee \text{CogSci}) \tag{2}$$
$$\text{ReadsDennett} \supset (\text{Phil} \vee \text{CogSci}) \tag{3}$$
$$\text{ReadsKosslyn} \supset (\text{Psych} \vee \text{CogSci}) \tag{4}$$

Clause (1) states a sufficient condition for being a cognitive scientist: being a computer scientist, *and* a philosopher, *and* a psychologist. The remaining clauses let one deduce a person's profession from his or her reading habits. Clause (2) states that if a person reads papers written by McCarthy, then the person is either a computer scientist or a cognitive scientist (or possibly both). Similarly, a reader of Dennett is either a philosopher or cognitive scientist or both, and a reader of Kosslyn is either a psychologist or cognitive scientist or both.

Reasoning with this theory can be quite complicated. For example, by reasoning by cases, one can prove that if Diane is a computer scientist who reads Dennett and Kosslyn, then Diane is a cognitive scientist. In general, for such non-Horn form theories, finding a proof may take time exponential in the length of the entire theory (provided $P \neq NP$).

Clause (1) can be resolved with subsets of clauses (2—4) to yield many different Horn clauses, such as

$$(\text{ReadsMcCarthy} \wedge \text{Phil} \wedge \text{ReadsKosslyn}) \supset \text{CogSci}$$
$$(\text{CompSci} \wedge \text{ReadsDennett} \wedge \text{Psych}) \supset \text{CogSci}$$

In fact, the LUB of this theory is equivalent to the set of 2^3 Horn clauses:

$$\left\{ \begin{array}{l} (p \wedge q \wedge r) \supset \text{CogSci} \quad | \\ \qquad p \in \{\text{CompSci}, \text{ReadsMcCarthy}\} \\ \qquad q \in \{\text{Phil},\ \text{ReadsDennett}\} \\ \qquad r \in \{\text{Psych},\ \text{ReadsKosslyn}\} \end{array} \right\} \tag{5}$$

Furthermore, we can prove that there is *no* smaller set of Horn clauses equivalent to (5). Note that this is a much stronger condition then simply saying that there are no redundant clauses in (5); we are asserting that there is no way to represent the same information in less space by using Horn clauses (or even using non-Horn clauses, for that matter). In general:

Theorem 1 *There exist clausal theories Σ of size n such that the smallest clausal representation of their LUB is of size $O(2^n)$.*

The proofs of this and all other theorems appear in the appendix. Thus, although we can tell if any clause follows from the LUB in time linear in the size of the LUB, the explosion in size of the LUB in this example wipes out our savings. Of course, there are also many commonsense theories for which such exponential blowup does *not* occur.

4 Shrinking the LUB

There are many ways to modify the syntactic characteristics of a theory without changing its basic meaning. For example, any theory can be represented by a set of clauses each containing no more than three literals (3-CNF form) by introducing new propositional letters. The old and new theories are not equivalent, since the new uses an expanded vocabulary, but they are essentially the same: they both entail or both do not entail any formula that does not contain any of the new letters.

Thus one might wonder if a large LUB could be represented by a small set of Horn clauses that have basically the same meaning, if not actual logical equivalence. As with the case of 3-CNF formulas, the technique we use depends on the introduction of new propositional letters. Rather than modify the definition of the LUB, we will add these new letters to the source theory itself. If we take the meaning of a letter to be a concept, we will see that the method reduces to the definition of new concepts that generalize old concepts.

For example, let us modify the theory given by clauses (1—4) by introducing three new concepts, "computer science buff", "philosophy buff", and "psychology buff". The first generalizes the concepts of a computer scientist and of a reader of papers by McCarthy. Similarly, the second generalizes philosopher and reader of Dennett, and the third generalizes psychologist and reader of Kosslyn. Each concept definition requires three clauses: one to assert that the more general concept is divided among its subconcepts, and two to assert that the subconcepts are part of the concept. The added clauses are:

$$\text{CompSciBuff} \supset (\text{CompSci} \vee \text{ReadsMcCarthy}) \tag{6}$$

$$\text{CompSci} \supset \text{CompSciBuff} \tag{7}$$

$$\text{ReadsMcCarthy} \supset \text{CompSciBuff} \tag{8}$$

$$\text{PhilBuff} \supset (\text{Phil} \vee \text{ReadsDennett}) \tag{9}$$

$$\text{Phil} \supset \text{PhilBuff} \tag{10}$$

$$\text{ReadsDennett} \supset \text{PhilBuff} \tag{11}$$

$$\text{PsychBuff} \supset (\text{Psych} \vee \text{ReadsKosslyn}) \tag{12}$$

$$\text{Psych} \supset \text{PsychBuff} \tag{13}$$

$$\text{ReadsKosslyn} \supset \text{PsychBuff} \tag{14}$$

The LUB of the augmented theory containing (1—4) and the clauses above can be represented by just the Horn clauses from the new concept definitions (7, 8, 10, 11, 13, 14) together with the single clause

$$(\text{CompSciBuff} \wedge \text{PhilBuff} \wedge \text{PsychBuff}) \supset \text{CogSci} \tag{15}$$

Returning to our example above where Diane is a computer scientist who reads Dennett and Kosslyn, we can now infer quickly from (7), (11), and (14) that she is a computer science buff, philosophy buff, and psychology buff, and therefore by (15) a cognitive scientist. Note that this inference can be computed in time linear in the size of the new LUB and therefore linear in the size of the *original* theory (1—4).

So, by teaching the system new concept definitions, the size of the new source theory grows only linearly, and the LUB shrinks to approximately the size of the source theory itself. [2]

5 Inducing New Concepts

So far we have seen that the goal of speeding inference by creating a compact, tractable approximation of a knowledge base can motivate learning concepts that are simple generalizations of previous concepts. This presupposes the existence of a helpful teacher and/or a separate concept learning module that will present the knowledge compilation system with useful concept definitions. One might wonder, however, if the process can be inverted: Can such concepts be generated as a by-product of the search for a compact representation of the tractable approximation? This is indeed possible, we will show below. (See [Muggleton and Buntine, 1988] for a different approach to learning new generalizations, based on inverting resolution proofs.)

Suppose you know that two different classes of objects, call them p and q, share a number of characteristics. For example, to represent the fact that all p's and all q's are blue or red or orange one could write

$$p \supset (\text{ blue } \vee \text{ red } \vee \text{ orange })$$
$$q \supset (\text{ blue } \vee \text{ red } \vee \text{ orange })$$

In such a situation it seems quite reasonable to hypothesize the existence of a class of objects r that subsumes both p and q, and which has the characteristic properties p and q share. That is, you would create a new symbol r, and add the definition $r \equiv (p \vee q)$ to your knowledge base. The common properties of p and q can be associated with this new concept r, and the original axioms stating those properties (namely the two clauses above) can be be deleted from your knowledge base, without loss of information. Thus the original axioms are replaced by

$$r \supset (\text{ blue } \vee \text{ red } \vee \text{ orange })$$
$$p \supset r$$
$$q \supset r$$

[2] The significance of this reduction does not lie solely in the fact that some *arbitrary* representation equivalent to the LUB can be encoded in a linear number of characters. For instance, the schema in equation (5) is also written using no more characters than there are literals in the original theory. Or even more to the point, the source theory itself can be taken to "represent" its own LUB, where we interpret it to mean "the set of all Horn clauses that can be derived from this set of formulas." The reason the reduction given in this example is interesting is that the resulting representation also allows efficient inference — that is, linear in the size of the representation.

The new axioms, even without the addition of the axiom $r \supset (p \vee q)$, have a number of desirable properties. We will state these properties for the general case.

Definition: Induced Concept
Let θ be a set of clauses containing one or more pairs of clauses of the form

$$\neg p \vee \alpha_i, \quad \neg q \vee \alpha_i \tag{16}$$

where p and q are letters and $\alpha_1, \cdots, \alpha_n$ for $n \geq 1$ are disjunctions of literals. An *induced concept* of θ is a new letter r together with two kinds of defining clauses: one for the *necessary* condition

$$r \supset (p \vee q) \tag{17}$$

and a pair of clauses for *sufficient* conditions:

$$p \supset r, \qquad q \supset r \tag{18}$$

Definition: Compaction
Let θ be a set of clauses, and r an induced concept of θ. Then θ', the *compaction* of θ using r, is defined as follows:

$$\begin{aligned} \theta' \quad = \quad \theta \quad & - \{\neg p \vee \alpha_i \mid i \in \{1, \cdots, n\}\} \\ & - \{\neg q \vee \alpha_i \mid i \in \{1, \cdots, n\}\} \\ & \cup \{\neg p \vee r, \quad \neg q \vee r\} \\ & \cup \{\neg r \vee \alpha_i \mid i \in \{1, \cdots, n\}\} \end{aligned}$$

That is, the compaction is obtained by removing the clauses given in (16), and adding the clauses given in (18), and adding a set of clauses that states that r implies each of the α_i.

Theorem 2 *Let θ' be a compaction of θ using induced concept r. Then*

- *If α is a formula not containing r, then $\theta \models \alpha$ if and only if $\theta' \models \alpha$. In other words, θ' is a conservative extension of θ.*
- *If the total length of the α_i's is 4 or more, then θ' is smaller than θ.*

Let us see what happens when the large LUB given by equation (5) is repeatedly compacted by inducing new concepts. Arranging the the clauses of the LUB as follows suggests that CompSci and ReadsMcCarthy should be generalized:

$$\begin{aligned} &(\text{CompSci} \wedge \text{Phil} \wedge \text{Psych}) \supset \text{CogSci} \\ &(\text{ReadsMcCarthy} \wedge \text{Phil} \wedge \text{Psych}) \supset \text{CogSci} \\ &(\text{CompSci} \wedge \text{ReadsDennett} \wedge \text{Psych}) \supset \text{CogSci} \\ &(\text{ReadsMcCarthy} \wedge \text{ReadsDennett} \wedge \text{Psych}) \supset \text{CogSci} \\ &(\text{CompSci} \wedge \text{Phil} \wedge \text{ReadsKosslyn}) \supset \text{CogSci} \\ &(\text{ReadsMcCarthy} \wedge \text{Phil} \wedge \text{ReadsKosslyn}) \supset \text{CogSci} \\ &(\text{CompSci} \wedge \text{ReadsDennett} \wedge \text{ReadsKosslyn}) \supset \text{CogSci} \\ &(\text{ReadsMcCarthy} \wedge \text{ReadsDennett} \wedge \text{ReadsKosslyn}) \\ &\qquad \supset \text{CogSci} \end{aligned}$$

So the LUB can be compacted by generating a new symbol (let us call it "CompSciBuff"), and rewriting it as

$$(\text{CompSciBuff} \wedge \text{Phil} \wedge \text{Psych}) \supset \text{CogSci}$$
$$(\text{CompSciBuff} \wedge \text{ReadsDennett} \wedge \text{Psych}) \supset \text{CogSci}$$
$$(\text{CompSciBuff} \wedge \text{Phil} \wedge \text{ReadsKosslyn}) \supset \text{CogSci}$$
$$(\text{CompSciBuff} \wedge \text{ReadsDennett} \wedge \text{ReadsKosslyn}) \supset \text{CogSci}$$
$$\text{CompSci} \supset \text{CompSciBuff}$$
$$\text{ReadsMcCarthy} \supset \text{CompSciBuff}$$

The pair of propositions Phil andReadsDennett fit the pattern of the concept induction rule, so we introduce a symbol called PhilBuff and rewrite again:

$$(\text{CompSciBuff} \wedge \text{PhilBuff} \wedge \text{Psych}) \supset \text{CogSci}$$
$$(\text{CompSciBuff} \wedge \text{PhilBuff} \wedge \text{ReadsKosslyn}) \supset \text{CogSci}$$
$$\text{CompSci} \supset \text{CompSciBuff}$$
$$\text{ReadsMcCarthy} \supset \text{CompSciBuff}$$
$$\text{Phil} \supset \text{PhilBuff}$$
$$\text{ReadsDennett} \supset \text{PhilBuff}$$

Finally, concept induction is applied to the pair Psych and ReadsKosslyn. The result is the small LUB presented at the end of the previous section:

$$(\text{CompSciBuff} \wedge \text{PhilBuff} \wedge \text{PsychBuff}) \supset \text{CogSci}$$
$$\text{CompSci} \supset \text{CompSciBuff}$$
$$\text{ReadsMcCarthy} \supset \text{CompSciBuff}$$
$$\text{Phil} \supset \text{PhilBuff}$$
$$\text{ReadsDennett} \supset \text{PhilBuff}$$
$$\text{Psych} \supset \text{PsychBuff}$$
$$\text{ReadsKosslyn} \supset \text{PsychBuff}$$

In general:

Theorem 3 *There exist clausal theories Σ of size n such that (1) the smallest clausal representation of their LUB is of size $O(2^n)$ and (2) there are compactions of their LUB (using one or more induced concepts) that are of size $O(n)$.*

Although we have been thinking of induction as a technique for reducing the size of the LUB, it can also be thought of as a method for adding new concepts to the source theory, as illustrated by the following theorem.

Theorem 4 *Let Σ be a set of clauses and Σ_{lub} its Horn least upper-bound. Let Σ'_{lub} be a compaction of Σ_{lub} using induced concept r. Define Γ to be Σ together with (both the necessary and sufficient conditions of) the definition of r:*

$$\Gamma = \Sigma \cup \{r \equiv (p \vee q)\}$$

Then the Horn least upper-bound Γ_{lub} *of* Γ *entails the compaction* Σ'_{lub}*:*

$$\Gamma_{\text{lub}} \models \Sigma'_{\text{lub}}$$

Furthermore, Γ_{lub} *is a conservative extension of* Σ_{lub}*.*

In other words, compacting the LUB by inducing new concepts can be viewed as a result of regenerating the LUB after adding the definition of a new concept to the source theory. In the example given in this paper, the compacted and regenerated LUB's are equivalent, but in general the regenerated LUB can be slightly stronger. For example, if the theory Σ is $\{p \supset s,\ q \supset s,\ t \supset (p \vee q)\}$, the reader may verify that Γ_{lub} entails $t \supset r$, but that Σ'_{lub} does not. In any case, the regenerated bound Γ_{lub} still does not entail any formulas not containing the new concept r that are not entailed by the original bound Σ_{lub}.

6 Do Efficient Representations Always Exist?

So far we have shown that a naive representation of a theory's LUB can sometimes require an exponential amount of space, and that in some of those cases a clever representation using new propositional letters requires only a polynomial amount of space. The question then becomes how general these results are. Is it always possible to produce a small representation of the LUB by the compaction technique described above? Failing that, one may wonder if it is always possible to produce a small and tractable representation of a theory's LUB using *any* techniques and data structures, including methods we have not yet discovered.

The following theorem states that the more general question is in fact equivalent to a major open question in complexity theory, whose answer is expected to be negative.

Theorem 5 *Unless NP* $\subseteq$ *non-uniform P, it is not the case that the Horn least upper bound* Σ_{lub} *of a propositional clausal theory* Σ *can always be represented by a data structure that allows queries of the form*

$$\Sigma_{\text{lub}} \models \alpha$$

to be answered in time polynominal in $(|\Sigma| + |\alpha|)$*, where* α *is a single Horn clause.*

Note that this is so despite the fact that we allow an arbitrary amount of work to be performed in computing the data structure used to represent Σ_{lub}.

The notion of "non-uniform P" comes from work in circuit complexity [Boppana and Sipser, 1990]. A problem is in non-uniform P (also called P/poly) iff for every integer n there exists a circuit of complexity (size) polynomial in n that solves instances of size n. The adjective "non-uniform" refers to the fact that different circuits may be required for different values of n. Any problem that can by solved by an algorithm that runs in time $O(f(n))$ has circuit complexity $O(f(n) \log n)$. We use this fact implicitly in the proof of the theorem, where we talk about polynomial time algorithms rather than polynomial size circuits.

The class non-uniform P is, however, considerably larger than P. (For example, non-uniform P contains non-computable functions, such as the function that returns "1" on inputs of length n iff Turing Machine number n halts on all inputs. For any n the circuit is simply fixed to return 1 or 0.) Although it is possible that P $\neq$ NP and yet NP $\subseteq$ non-uniform P, this is considered unlikely. One consequence would be that the polynomial-time hierarchy would collapse to Σ_2 [Karp and Lipton, 1982]. As shown in the appendix, the theorem can be strengthened to say that the claim that there always exists an efficient form of the LUB for answering Horn clausal queries is *equivalent* to the claim that NP $\subseteq$ non-uniform P. Therefore a proof that such efficient representations do or do not always exist would be a major result in the complexity of finite functions.

An immediate corollary of the theorem is that unless the polynomial hierarchy collapses in this manner, compaction by defining new propositions is an incompletement method for shrinking the LUB.

Corollary *Unless NP $\subseteq$ non-uniform P, it is not the case that there is always a compaction (using any number of new variables) of the Horn least upper bound of a theory Σ that is of size polynomial in $|\Sigma|$.*

This follows from the theorem because one can determine if a Horn clause follows from a compaction (which is itself Horn) in time linear in the length of the compaction plus the length of the query.

7 Conclusions

Knowledge compilation speeds inference by creating two tractable approximations of a knowledge base. Part of the potential speed advantage can be lost if one of these bounds, the LUB, grows to exponential size. We have shown that this can sometimes occur, but that learning defined concepts can sometimes rescue the method, by allowing an exponential reduction in the size of the LUB. In addition, compacting the tractable theory by using a rule of induction is one way to generate the concepts.

These results provide a useful bridge between the areas of concept learning and speed-up learning. While work in concept learning is normally motivated by the need to classify data, our work suggests that concept learning may also be useful for efficient commonsense reasoning.

Finally, we prove that unless a radical reformulation of complexity theory occurs it is likely that any method for efficiently representing the Horn least upper-bound is incomplete. However, the general framework for theory approximation described in the beginning of this paper can be adapted to use bounds that are in any tractable language, such as binary clauses [Kautz and Selman, 1991], or Horn clauses of a fixed maximum length. In some of these languages we can be sure that the LUB is of polynomial size. For example, the binary-clause least upper-bound of a theory of size n is of size $O(n^2)$. Currently we are studying the properties of these different kinds of approximations.

Appendix: Proofs

Proof of Theorem 1

Consider a theory Σ of the following form, for arbitrary n:

$$\begin{array}{l}
\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n \vee s \\
\neg q_1 \vee p_1 \vee s \\
\neg q_2 \vee p_2 \vee s \\
\vdots \\
\neg q_n \vee p_n \vee s
\end{array}$$

We see that Σ contains $4n+1$ literals; that is, it is of size $O(n)$. The LUB Σ_{lub} of Σ is the following set of Horn clauses, of total length 2^n:

$$\left\{\begin{matrix}\neg x_1 \vee \neg x_2 \vee \cdots \vee \neg x_n \vee s \mid \\ x_i \in \{p_i, q_i\} \text{ for } 1 \leq j \leq n\end{matrix}\right\}$$

First we prove that the set Σ_{lub} has the following properties: no two clauses resolve, and it is irredundant (no subset of Σ_{lub} implies all of Σ_{lub}). Then we prove that Σ_{lub} is of minimum size. (Note that being of minimum size is a stronger condition than being irredundant.)

Proof that Σ_{lub} is irredundant: suppose there is a clause α in Σ_{lub} such that $\Sigma_{\text{lub}} - \{\alpha\} \models \alpha$. Since no two clauses in $\Sigma_{\text{lub}} - \{\alpha\}$ resolve, by completeness of resolution there must be an α' in $\Sigma_{\text{lub}} - \{\alpha\}$ such that α' subsumes α. But this is impossible, since all clauses in Σ_{lub} are of the same length.

Finally, we can now prove that there is no smaller set of clauses Σ'_{lub} which is equivalent to Σ_{lub}: Suppose there *were* an Σ'_{lub} such that $\Sigma_{\text{lub}} \equiv \Sigma'_{\text{lub}}$ and $|\Sigma'_{\text{lub}}| < |\Sigma_{\text{lub}}|$. Then for all α in Σ'_{lub},

$$\Sigma_{\text{lub}} \models \alpha$$

and since no clauses in Σ_{lub} resolve, this means that there exists an α' in Σ_{lub} such that α' subsumes α.

That is, every clause in Σ'_{lub} is subsumed by some clause in Σ_{lub}. Suppose each clause in Σ_{lub} subsumed a different clause in Σ'_{lub}; then $|\Sigma_{\text{lub}}'| \geq |\Sigma_{\text{lub}}|$, a contradiction. Therefore there is a proper subset Σ_{lub}'' of Σ_{lub} such that each clause in Σ'_{lub} is subsumed by some clause in Σ_{lub}''.

Then $\Sigma_{\text{lub}}'' \models \Sigma_{\text{lub}}'$, and therefore $\Sigma_{\text{lub}}'' \models \Sigma_{\text{lub}}$. But this is impossible, because we saw that Σ_{lub} is irredundant. Therefore there is no smaller set of clauses equivalent to Σ_{lub} which is shorter than Σ_{lub}. $\Box$

Proof of Theorem 2

(First part) Let $\theta'' = \theta \cup \{r \equiv (p \vee q)\}$. We see that θ'' is a conservative extension of θ, since it extends θ by adding a defined proposition [Shoenfield, 1967, page 57]. Since $\theta' \models \theta$ it is an extension of θ, and since $\theta'' \models \theta'$ it must be the case that θ' is a conservative extension of θ.

(Second part) Obvious, by counting. $\Box$

Proof of Theorem 3

The LUB of any theory in the class described in the proof of Theorem 1 has the following compaction of size $O(n)$ using new letters $r_1, \ldots, r_n$:

$$\begin{array}{l}\{\neg r_1 \vee \cdots \vee \neg r_n \vee s\} \cup \\ \{\neg p_i \vee r_i, \quad \neg q_i \vee r_i \mid 1 \leq i \leq n\}\end{array} \quad \Box$$

Proof of Theorem 4

(First part, that $\Gamma_{\text{lub}} \models \Sigma_{\text{lub}}'$.) Observe that

$$\Sigma \cup \{r \equiv (p \vee q)\} \models \{r \supset \alpha_i | i \in \{1, ..., n\}\}.$$

Together with the fact that $\Sigma \models \Sigma_{\text{lub}}$ it follows that

$$\Sigma \cup \{r \equiv (p \vee q)\} \models \Sigma_{\text{lub}} \cup \{r \supset \alpha_i | i \in \{1, ..., n\}\}$$

and thus $\Gamma_{\text{lub}} \models \Sigma_{\text{lub}}'$.

(Second part, that $\Gamma_{\rm lub}$ is a conservative extension of $\Sigma_{\rm lub}$.) It is plain that $\Gamma_{\rm lub}$ is an extension of $\Sigma_{\rm lub}$, because adding premises to the source theory can only make the LUB grow monotonically. Next we prove that $\Gamma_{\rm lub}$ is conservative. Let α be a formula not containing r such that $\Gamma_{\rm lub} \models \alpha$. Let us first consider the case where α is a clause and not a tautology. Then there is a Horn clause α' such that

$$\Gamma_{\rm lub} \models \alpha' \text{ and } \alpha' \models \alpha$$

and therefore $\Gamma \models \alpha'$. Recall that Γ is a conservative extension of Σ, and α is in language of Σ. This means that $\Sigma \models \alpha'$, and since α' is Horn, $\Sigma_{\rm lub} \models \alpha'$ and therefore $\Sigma_{\rm lub} \models \alpha$. For the general case, note that any α can be equivalently written as a conjunction of clauses, and $\Gamma_{\rm lub}$ thus entails each clause. Thus $\Sigma_{\rm lub}$ entails each clause, and therefore entails α. □

Proof of Theorem 5

Suppose such a representation of the LUB always existed. We then show that 3-SAT over n variables can be determined in $O(n^3)$ time. Because the choice of n is arbitrary, and 3-SAT is an NP-complete problem, this would mean that NP $\subseteq$ non-uniform P.

Let the variables be a set of main variables $\{p_1, \ldots, p_n\}$ together with a set of auxiliary variables

$$\{c_{x,y,z} | x, y, z \in \text{LITS}\}.$$

where LITS is the set of literals constructed from the main variables (the variables or their negations). Let the source theory Σ be the conjunction of clauses:

$$\Sigma = \bigwedge_{x,y,z \in \text{LITS}} \{x \vee y \vee z \vee \neg c_{x,y,z}\}$$

Note that Σ is of length $O(n^3)$. The idea behind the construction is that any 3-SAT clause over n variables can be constructed by selecting a subset of the clauses from Σ and eliminating the auxiliary variables.

Now suppose we have been able to compute a representation of the Horn LUB of Σ that has the property that one can test Horn queries against it in polynomial time. We noted before that a Horn formula follows from the LUB if and only if it follows from the source theory.

Suppose Δ is an arbitrary 3-CNF formula over n variables that we wish to test for satisfiability. We construct a Horn clause α containing only auxiliary variables by including a negative auxiliary variable c that corresponds to each clause δ in Δ. For example, if Δ is

$$(p_1 \vee \neg p_3 \vee p_5) \wedge (\neg p_1 \vee p_2 \vee \neg p_4)$$

then the corresponding Horn clause is

$$\neg c_{p_1, \neg p_3, p_5} \vee \neg c_{\neg p_1, p_2, \neg p_4}$$

Now we claim that this Horn clause is implied by the LUB if and only if Δ is not satisfiable.

($\rightarrow$) Suppose the query is implied by the LUB. Since the query is Horn, this means that

$$\Sigma \models \neg c \vee \neg c' \vee \neg c'' \vee \ldots$$

where the $\{c, c', c'', \ldots\}$ are the auxiliary variables in the query that correspond to clauses $\{\delta, \delta', \delta'', \ldots\}$ in Δ. Equivalently,

$$\Sigma \cup \{\neg(\neg c \vee \neg c' \vee \neg c'' \vee \ldots)\} \text{ is unsatisfiable.}$$

That is,

$$\Sigma \cup \{c, c', c'', \ldots\} \text{ is unsatisfiable.}$$

Note that any clause in Σ containing an auxiliary variable other than $\{c, c', c'' \ldots\}$ can be eliminated, since that clause is satisfied in any model in which its auxiliary variable is assigned false, and no other instance of that variable appears in the formula. Thus it must be the case that

$$\{\delta \vee \neg c, \delta' \vee \neg c', \delta'' \vee \neg c'', \ldots\} \cup \{c, c', c'', \ldots\}$$

is unsatisfiable. Because the auxiliary variables each appear exactly once negatively and once positively above, they can can be resolved away. Therefore

$$\{\delta, \delta', \delta'', \ldots\} \equiv \Delta \text{ is unsatisfiable.}$$

($\leftarrow$) Note that each step in the previous section can be reversed, to go from the assumption that Δ is unsatisfiable to the conclusion that $\Sigma \models \alpha$.

We assumed that the test could be peformed in time polynominal in the length of the source theory plus the length of the query. We noted earlier that the source theory is of length $O(n^3)$. The query is also of of length $O(n^3)$, because there are only n^3 auxiliary variables. The smallest Δ containing n variables is of length n, so in in any case both the source theory and query are polynomial in the length of Δ. Thus satisfiability of Δ can be determined in time polynomial in the length of Δ. Since the choice of n was arbitrary, and 3-SAT is an NP-complete problem, this means that NP $\subseteq$ non-uniform P. $\Box$

Proof of strengthened Theorem 5

We can strengthen the theorem to an equivalence by showing that NP $\subseteq$ non-uniform P implies that small and tractable representations of the LUB always exist. Suppose we are given a source theory Σ of length m containing n variables. Assuming NP $\subseteq$ non-uniform P, there exists a circuit that determines satisfiability of formulas of length $m + n$ that has complexity polynomial in $m + n$. We use this circuit to construct program to test queries of the form $\Sigma_{\text{lub}} \models \alpha$ as follows: given α, first check that is not a tautology, and eliminate any duplicated literals. The resulting query is of length $\leq n$. Then pad out the query to exactly length n by duplicating any of its literals. Then the negation of the query together with Σ is a formula of exactly length $m + n$, so we can use the circuit to determine if the formula is unsatisfiable, or equivalently, that α follows from Σ. Since α is Horn, then the latter condition is equivalent to saying $\Sigma_{\text{lub}} \models \alpha$. Since the circuit is of size polynomial in $m + n$ it must execute in time polynomial in $m + n$ — that is, in time polynomial in $(|\Sigma| + |\alpha|)$. $\Box$

References

[Amarel, 1968] Saul Amarel. On representations of problems of reasoning about actions. In Michie, editor, *Machine Intelligence 3*, pages 131–171. Edinburgh University Press, 1968.

[Boppana and Sipser, 1990] R. B. Boppana and M. Sipser. The complexity of finite functions. In J. an Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Compexity*, pages 757–804. Elsevier, Amsterdam (and MIT Press, Cambridge), 1990.

[Bresina *et al.*, 1986] J. L. Bresina, S. C. Marsella, and C. F. Schmidt. Reappr - improving planning efficience via local expertise and reformulation. Technical Report LCSR-TR-82, Rutgers U., New Brunswick, June 1986.

[Cook, 1971] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, pages 151–158, 1971.

[Dowling and Gallier, 1984] William F. Dowling and Jean H. Gallier. Linear time algorithms for testing the satisfiability of propositional horn formula. *Journal of Logic Programming*, 3:267–284, 1984.

[Greiner, 1992] Russell Greiner. Learning near-optimal horn approximations. In *Preprints of the AAAI Spring Symposium on Knowledge Assimilation*. Stanford University, Stanford, CA, March 1992.

[Karp and Lipton, 1982] R. M. Karp and R. Lipton. Turing machines that take advice. *Enseign. Math.*, 28:191–209, 1982.

[Kautz and Selman, 1991] Henry Kautz and Bart Selman. A general framework for knowledge compilation. In *Proceedings of the International Workshop on Processing Declarative Knowledge (PDK)*, Kaiserslautern, Germany, July 1991.

[Kearns, 1990] Michael J. Kearns. *The Computational Complexity of Machine Learning*. MIT Press, Boston, MA, 1990.

[Minton, 1985] Steve Minton. Selectively generalizing plans for problem solving. In *Proceedings of IJCAI-85*, 1985.

[Minton, 1988] Steven Minton. *Learning Effective Search Control Knowledge: an Explanation-Based Approach*. PhD thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1988.

[Muggleton and Buntine, 1988] Stephen Muggleton and Wray Buntine. Machine invention of first-order predicates by inverting resolution. In J. Laird, editor, *Proceedings of the Fifth International Conference on Machine Learning*, page 339, Los Altos, CA, 1988. Morgan Kaufmann.

[Pagallo and Haussler, 1990] Giulia Pagallo and David Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, 5:77–99, 1990.

[Selman and Kautz, 1991] Bart Selman and Henry Kautz. Knowledge compilation using horn approximations. In *Proceedings of AAAI-91*, pages 904–909, Anaheim, CA, 1991.

[Shoenfield, 1967] Joseph R. Shoenfield. *Mathematical Logic*. Addison-Wesley, Reading, MA, 1967.

[Subramanian and Genesereth, 1987] Devika Subramanian and Michael R. Genesereth. The relevance of irrelevance. In *Proceedings of IJCAI-87*, volume 1, page 416, 1987.