A Scalable Weighted Max-SAT Implementation of Propositional Etcetera Abduction

Naoya Inoue
Graduate School of Information Sciences
Tohoku University
naoya-i@ecei.tohoku.ac.jp

Andrew S. Gordon
Institute for Creative Technologies
University of Southern California
gordon@ict.usc.edu

Abstract

Recent advances in technology for abductive reasoning, or inference to the best explanation, encourage the application of abduction to real-life commonsense reasoning problems. This paper describes *Etcetera Abduction*, a new implementation of logical abduction that is both grounded in probability theory and optimized using contemporary linear programming solvers. We present a Weighted Max-SAT formulation of Etcetera Abduction, which allows us to exploit highly advanced technologies developed in the field of SAT and Operations Research. Our experiments demonstrate the scalability of our proposal on a large-scale synthetic benchmark that contains up to ten thousand axioms, using one of the state-of-the-art mathematical optimizers developed in these fields. This is the first work to evaluate a SAT-based approach to abductive reasoning at this scale. The inference engine we developed has been made publicly available.

Introduction

Logical abduction is a form of automated reasoning that searches for hypotheses that, if they were true, would logically entail a set of input observations given a knowledge base of inference rules. Unlike logical theorem-proving, where the task is to identify the truth value of a logical sentence, logical abduction is well suited to artificial intelligence problems of explanation and commonsense reasoning, i.e., where the search is for the best set of assumptions that explain what we observe given what we know. Hobbs et al. (1993) proposed "*interpretation as abduction*," casting the natural language understanding problem in terms of logical abduction, where the search is for the best representation that accounts for the observable text. Others have applied logical abduction to non-textual interpretation problems, e.g., the interpretation of agent behavior (Meadows, Heald, and Langley 2015; Gordon 2016).

Automating the process of logical abduction requires both generating candidate hypotheses and ranking hypotheses according to their quality. Recent research has seen substantial improvements in the efficiency and scalability of cost-based logical abduction, where the combinatorial search is managed by optimized linear programming solvers (Inoue and Inui 2014; Yamamoto et al. 2015). In parallel, others have pursued probabilistic formulations of logical abduction based on uncertain inference (Blythe et al. 2011; Ovchinnikova, Gordon, and Hobbs 2013).

In this paper, we describe a new implementation of logical abduction that is both grounded in probability theory and optimized using contemporary linear programming solvers. This solution, named *Etcetera Abduction*, combines ideas from two previous formulations: Weighted Abduction (Hobbs et al. 1993) and Probabilistic Horn Abduction (Poole 1993). We provide a SAT formulation of Etcetera Abduction for propositional logic that enables abductive reasoning problems to be addressed using a weighted maximum satisfiability solver (Weighted Max-SAT), with a straightforward translation as an Integer Linear Programming (ILP) problem. Using a high-performance ILP solver, we evaluated our approach compared to a reference implementation of Etcetera Abduction, using synthetic knowledge bases consisting of hundreds of thousands of axioms.

Background

Abduction

Abduction is inference to the best hypothesis (or explanation). In this paper, we assume propositional logic as the meaning representation of abduction. Abduction on propositional logic (henceforth, *propositional abduction*) is generally defined as follows:

- **Given:** (i) Background knowledge B, (ii) observations O, (iii) set A of propositional atoms, and (iv) evaluation function $eval$, where B is a set of propositional logic formulae, and O is a set of propositional literals.
- **Find:** Among a set $\mathcal{H}$ of hypotheses, where $\mathcal{H} \equiv \{H \subseteq A \mid H \cup B \models O, H \cup B \not\models \perp\}$, find the best hypothesis $H^* \in \mathcal{H}$ that maximizes $eval(H^*)$ (i.e., $H^* = \arg\max_{H \in \mathcal{H}} eval(H)$).

We refer to $a \in A$ as an *abducible*, $H \in \mathcal{H}$ as a *candidate hypothesis*, and $h \in H$ as an *elemental hypothesis*.

To model $eval$, a wide variety of methods have been proposed in the literature, ranging from a probabilistic measure to a cost-based measure (Charniak and Goldman 1991; Hobbs et al. 1993; Poole 1993, etc.). The probabilistic measure used in this paper is described in the next section.

Etcetera Abduction

Etcetera Abduction, introduced by Gordon (2016), provides a means of ordering abductive proofs by the probability of their assumptions, while adhering to strict notions of logical entailment, rather than uncertain inference. In terms of probability theory, abduction can be viewed as a Maximum A Posteriori (MAP) estimation for which we find the most likely hypothesis given the input observations:

$$\arg\max_{H \in \mathcal{H}} eval(H) = Pr(H|O) = \frac{Pr(O|H)Pr(H)}{Pr(O)} \quad (1)$$

In abduction, this maximization problem can be simply reduced to $\arg\max_{H \in \mathcal{H}} Pr(H)$ because H logically entails O (i.e., $Pr(O|H) = 1$) and $Pr(O)$ is constant in the maximization problem.

The approach of Etcetera Abduction expands on Poole (1993)'s Probabilistic Horn Abduction, where abduction proceeds by backchaining from observables by using axioms of Horn clauses, and where the joint probability $Pr(H)$ of any set of assumptions that logically entail the observables is naïvely calculated as the product of their priors, i.e., by assuming conditional independence over elemental hypotheses $h \in H$:

$$Pr(H) = \prod_{h \in H} Pr(h) \quad (2)$$

Etcetera Abduction adds to this idea by introducing a means of authoring defeasible axioms with clear probabilistic semantics. Following Hobbs et al. (1993), special literals are included in the antecedents of defeasible axioms, called *etcetera literals*, that represent all of the unspecified conditions that must also hold for the axiom to be logically true.

$$happy \rightarrow smile \qquad \text{not always true.} \quad (3)$$

$$happy \wedge etc_n \rightarrow smile \qquad \text{always true!} \quad (4)$$

These etcetera literals are specific to a single axiom in a knowledge base, and are here uniquely identified by a subscript number. Appearing only in antecedents, their truth cannot be proved, only assumed via logical abduction. They are related to McCarthy's abnormal predicates (*Ab*), used in his proposal for circumscription as a means of commonsense reasoning (McCarthy 1980), except with positive semantics. From the Latin "et cetera," meaning "and other things," etcetera literals stand in for all of the other conditions of the universe that would have to also be true for the remainder of the antecedent to logically imply the consequent. Formally, an etcetera literal E for a definite clause $A \wedge E \rightarrow C$ is defined as a disjunction of all possible conjunctions e where $A \wedge e \rightarrow C$, such that $A \wedge C \rightarrow E$. Or informally, whenever we have both A and C, the other conditions must have also been right for them both to be true.

In this definition, $(A \wedge C)$ is true exactly when $(A \wedge E)$ is true, giving us a means of specifying probabilistic semantics for etcetera literals. In situations where truth values represent the occurrence or non-occurrence of events, we have this equality between their joint probabilities.

$$Pr(A, E) = Pr(A, C) \quad (5)$$

Following Poole (1993), we assume etcetera literals are conditionally independent from all other literals, such that:

$$Pr(A)Pr(E) = Pr(A, C) \quad (6)$$

Solving for $Pr(E)$ gives us a conditional probability:

$$Pr(E) = Pr(C|A) \quad (7)$$

That is, the prior probability of an etcetera literal is equal to the conditional probability of the consequent given the rest of the antecedent. Likewise, when an etcetera literal E is the only antecedent for a consequent C, as in the axiom $E \rightarrow C$, then the prior probability of the etcetera literal E is equal to the prior probability of the consequent C.

In Etcetera Abduction, a knowledge base of definite clauses is constructed such that every axiom includes a unique etcetera literal in its antecedent, and every non-etcetera literal is a consequent in an axiom where a solitary etcetera literal is its antecedent. Each of these etcetera literals is assigned a probability, encoding the prior and conditional probabilities in the domain. Abduction proceeds by backchaining from input observations using this knowledge base to identify sets of assumptions consisting only of etcetera literals that logically entail the observations, and then ordering these sets by their joint probability assuming conditional independence (the product of their priors).

Etcetera Abduction has several advantages over other popular frameworks for logical abduction, most notably Weighted Abduction as proposed by Hobbs et al. (1993). Weighted Abduction proceeds by propagating the cost of leaving input observations unexplained back through a knowledge base of axioms, where each antecedent literal is annotated with a weight, a multiplier that pays the cost of the consequent with the assumption of the antecedent literal. Ovchinnikova, Gordon, and Hobbs (2013) showed that this cost propagation mechanism cannot be interpreted in terms of probabilities, and proposed instead an approach based on uncertain inference using Bayesian Networks. Etcetera Abduction takes a different approach, retaining Poole (1993)'s notion of strict logical entailment of input observations (rather than uncertain inference) by incorporating Hobbs et al. (1993)'s idea for etcetera literals, and by providing the probabilistic semantics for etcetera literals that affords a simple means of ordering sets of entailing assumptions.

However, technologies for the implementation of Weighted Abduction have improved dramatically in recent years. Inoue and Inui (2014) demonstrated that Weighted Abduction could be cast as an Integer Linear Programming problem, allowing the combinatorial search space to be efficiently explored using contemporary ILP solvers. For practical applications, Etcetera Abductions needs a comparably efficient implementation.

SAT-based Propositional Etcetera Abduction

Abductive reasoning is a constrained combinatorial optimization problem over abducibles. Given an abduction problem $\langle B, O, A, eval \rangle$, the task is to find the best hypothesis among a prohibitively large number of candidate hypotheses, where the number of candidate hypotheses is expressed by $O(2^{|A|})$. A propositional Etcetera Abduction problem is a special case of an abduction problem where A is a set of etcetera literals in B and $eval$ is $\prod_{h \in H} Pr(h)$. Unfortunately, the exponentially large search space prevents us from relying on exhaustive combinatorial search.

To address this issue, we show how to exploit a state-of-the-art mathematical optimizer developed in Operations Research for Etcetera Abduction. Specifically, we show how to recast an Etcetera Abduction problem as an ILP problem, preserving the correctness of the translation. We first show how to represent an abduction problem as a Weighted Max-SAT problem, and then present a clear translation from a Weighted Max-SAT problem into an ILP problem.

Weighted Max-SAT formulation

We begin with representing a purely logical abduction problem as a SAT problem, where the preference of hypothesis is not considered. The core idea is as follows. Let $\mathcal{A}$ be a truth assignment over abducibles A, and ϕ_{sc} be a logical formula that represents a sufficient condition of $\mathcal{A}$ entailing a disjunction of candidate hypotheses, i.e., $\phi_{sc} \models \bigvee_{H \in \mathcal{H}} H$. An abduction problem is then reduced to finding a truth assignment $\mathcal{A}$ that satisfies ϕ_{sc}, i.e., a SAT problem.

The important question is how to represent ϕ_{sc}. Console, Dupre, and Torasso (1991) formally discuss the relationship between abduction and deduction through Clark (1978)'s predicate completion. The Clark completion is a transformation procedure of background knowledge, assuming that all possible reasons are completely described in the background knowledge. The key result is that an abduction problem can be reduced to a deduction problem if Clark-completed background knowledge *with respect to non-abducible predicates* and input observations are given as premises. Let $comp_{na}(B)$ be a knowledge base obtained by the Clark completion of non-abducible predicates in B. They showed that the following relationship holds:

$$comp_{na}(B) \cup O \models \bigvee_{H \in \mathcal{H}} H \tag{8}$$

We leverage this key result as the sufficient condition ϕ_{sc}:

$$\phi_{sc} = comp_{na}(B) \wedge O \tag{9}$$

Following Console, Dupre, and Torasso (1991), we assume B to be a set of the following clauses:

$$h \leftarrow b_1 \wedge b_2 \wedge ... \wedge b_n (n \geq 0), \tag{10}$$

where h is an atom termed the *head* and each b_i is a literal. The literal is either an atom or a negated atom. We require B to be "hierarchical" (Clark 1978) and abducibles not to appear in the head of any clauses in B and input observations.

Consider the following Etcetera Abduction problem:

$$\begin{aligned} B = \{ & p \leftarrow a \wedge etc_{pa}, p \leftarrow b \wedge etc_{pb}, \\ & q \leftarrow a \wedge etc_{qa}, q \leftarrow c \wedge etc_{qc}, \\ & a \leftarrow etc_a, b \leftarrow etc_b, c \leftarrow etc_c\} \qquad (11) \\ O = \ & p \wedge q \qquad (12) \end{aligned}$$

ϕ_{sc} is then $comp_{na}(B) \wedge p \wedge q$, where:

$$\begin{aligned} comp_{na}(B) = \{ & p \leftrightarrow (a \wedge etc_{pa}) \vee (b \wedge etc_{pb}), \\ & q \leftrightarrow (a \wedge etc_{qa}) \vee (c \wedge etc_{qc}), \\ & a \leftrightarrow etc_a, b \leftrightarrow etc_b, c \leftrightarrow etc_c\} \end{aligned} \tag{13}$$

We can easily see that any truth assignment satisfying ϕ_{sc} entails a disjunction of candidate hypotheses (i.e., $\{etc_a, etc_{pa}, etc_{qa}\} \vee \{etc_a, etc_{pa}, etc_c, etc_{qc}\} \vee ...$).

We introduce *relevant reasoning* as a technique to reduce ϕ_{sc}. Recall that the search in Etcetera Abduction is over a set of candidate hypotheses that *logically entail* input observations. Thus, it is sufficient to include axioms relevant to input observations in ϕ_{sc}. Imagine the same abduction problem as that mentioned above except $O = p$. It is not necessary to include $c \leftrightarrow etc_c$ in $comp_{na}(B)$ because any truth assignment satisfying the remaining axioms already entails the disjunction of candidate hypotheses. Therefore, instead of computing $comp_{na}(B)$, we compute $comp_{na}(B')$, where B' is a minimal knowledge base B' such that $B' \subseteq B, B' \cup H \models O$. We obtain B' by collecting a set of axioms including literals reachable from input observations O by backward chaining.

Representing evaluation function. We extend the SAT problem as a Weighted Max-SAT problem. Given a weighted logical formula F (e.g., $p^{23.0} \wedge (p \rightarrow q)^{0.8}$), the Weighted Max-SAT problem is to find the best truth assignment for atoms in F that satisfies F, where the goodness is defined by the sum of weights of satisfied clauses. We represent the evaluation function $Pr(H) = \prod_{h \in H} Pr(h)$ of Etcetera Abduction by introducing a weighted logical formula ϕ_{ev} as follows. For each $a \in A$, we add a weighted clause to state that the inclusion of a into a hypothesis contributes to the satisfiability by $\log Pr(a)$:

$$\phi_{ev} = \bigwedge_{a \in A} a^{\log Pr(a)} \tag{14}$$

We then solve the Weighted Max-SAT problem of $\phi_{sc} \wedge \phi_{ev}$. Clearly, this is equivalent to finding the best hypothesis H that maximizes the *log* of $Pr(H)$.

ILP translation

We can use an arbitrary off-the-shelf solver for solving the Weighted Max-SAT problem. In this paper, we leverage the state-of-the-art technology of ILP, which has already been proven to be efficient for solving the Weighted Max-SAT problem (Ansótegui and Gabàs 2013, etc.).

To represent the Weighted Max-SAT problem as a linear programming problem, we first pre-process the translated formula ϕ_{sc} to represent the Weighted Max-SAT problem as a linear programming problem. One approach is to convert ϕ_{sc} into a conjunctive normal form (CNF) with De Morgan's law, where each clause can be represented by a linear constraint. However, it is well known that this conversion yields a CNF with an exponentially large number of clauses. We thus exploit the *Tseytin transformation* (Tseytin 1983). Given a logical formula ϕ, we first apply the following procedures for each subformula F in ϕ: (i) introduce a new variable x, (ii) replace F with x, and (iii) add $x \leftrightarrow F$ to ϕ. In terms of an Etcetera Abduction problem, given n explanations with m literals for q (i.e., $q \leftrightarrow \bigvee_i^n (l_{i,1} \wedge l_{i,2} \wedge ... \wedge l_{i,m})$), it generates $1 + n$ clauses (i.e. $q \leftrightarrow \bigvee_i^n x_i$ and for all $i \in \{1...n\}$, $x_i \leftrightarrow l_{i,1} \wedge l_{i,2} \wedge ... \wedge l_{i,m}$), whereas the naïve CNF conversion generates $n + m^n$ clauses. For example, given $\phi = p \wedge q \wedge ((p \leftrightarrow etc_p) \wedge (q \leftrightarrow etc_q \vee (a \wedge etc_{qa})) \wedge (a \leftrightarrow etc_a))$, $T(\phi) = x_1 \wedge (x_1 \leftrightarrow p \wedge q \wedge x_2 \wedge x_3 \wedge x_4) \wedge (x_2 \leftrightarrow (etc_p \leftrightarrow p)) \wedge (x_3 \leftrightarrow (x_{3,1} \leftrightarrow q)) \wedge (x_{3,1} \leftrightarrow etc_q \vee x_{3,2}) \wedge (x_{3,2} \leftrightarrow a \wedge etc_{qa}) \wedge (x_4 \leftrightarrow (etc_a \leftrightarrow a))$. We then directly map each clause in $T(\phi)$ into a linear constraint as follows.

ILP variables. To represent a truth assignment $\mathcal{A}$ to atoms $a_1, a_2, ..., a_n$ appearing in $T(\phi_{sc})$, we introduce binary ILP variables $s_{a_1}, s_{a_2}, ..., s_{a_n} \in \{0, 1\}$. The assignment $0, 1$ indicates a truth assignment of "false" or "true," respectively.

For notational convenience, we define $\delta(l)$ as follows:

$$\delta(l) = \begin{cases} s_a & \text{if } l \equiv a \\ 1 - s_a & \text{if } l \equiv \neg a, \end{cases} \tag{15}$$

where l is a literal and a is an atom. That is, $\delta(l)$ returns 1 if l is satisfied by a truth assignment $\mathcal{A}$; otherwise, 0.

ILP constraints. For each clause C of $T(\phi_{sc})$, we introduce ILP constraints according to the following rules.

- If $C \equiv l$, then introduce $\delta(l) = 1$.
- If $C \equiv l \leftrightarrow l_1 \wedge l_2 \wedge ... \wedge l_n$, then introduce $n \cdot \delta(l) \leq \sum_{i=1}^n \delta(l_i)$ and $\delta(l) \geq \sum_{i=1}^n \delta(l_i) - n + 1$.
- If $C \equiv l \leftrightarrow l_1 \vee l_2 \vee ... \vee l_n$, then introduce $\delta(l) \leq \sum_{i=1}^n \delta(l_i)$ and $n \cdot \delta(l) \geq \sum_{i=1}^n \delta(l_i)$.
- If $C \equiv l \leftrightarrow (l_1 \leftrightarrow l_2)$, then introduce (i) $\delta(l) \leq 1 - \delta(l_1) + \delta(l_2)$, (ii) $\delta(l) \leq \delta(l_1) + 1 - \delta(l_2)$, (iii) $2\delta(l) \geq 1 - \delta(l_1) + \delta(l_2)$, and (iv) $2\delta(l) \geq \delta(l_1) + 1 - \delta(l_2)$.

One can easily see the equivalence between the logical formulae and their corresponding linear constraints.

ILP objective. We represent the satisfiability of a truth assignment as an ILP objective function:

$$\max \sum_{a^w \in \phi_{ev}} w \cdot s_a \tag{16}$$

Evaluation

We evaluated the scalability of the proposed method on one hundred synthetic benchmark problems. We compared our system with ETCABDUCTIONPY, the Python-based reference implementation of Etcetera Abduction (Gordon 2016)[1]. ETCABDUCTIONPY uses exhaustive combinatorial search to find the best hypothesis. For fair comparison, we extended ETCABDUCTIONPY to implement our system so that basic modules such as the logical form parser are shared in both systems. We used the Gurobi Optimizer 7.0[2], a state-of-the-art commercial mathematical programming solver. In this experiment, we used a machine with Xeon E5-2430 v2 (2.5GHz) x 2 (12 core) and 96.0 GB memory. To simulate a practical situation, the systems were given a 60 second time limit and a 6.0 GB memory limit. The proposed system has been made publicly available at `https://github.com/naoya-i/EtcAbductionPy`.

	SMALLBENCH				
	Nodes	Edges	Axioms	Branch	Obs.
Min.	33	25	52	1.49	6
Max.	69	61	118	1.78	12
Avg.	54.4	45.9	89.0	1.6	9.0
	LARGEBENCH				
	Nodes	Edges	Axioms	Branch	Obs.
Min.	2,518	3,667	7,460	1.62	6
Max.	37,741	290,278	134,753	26.39	12
Avg.	20,232.9	156,024.3	74,123.8	4.9	8.9

Table 1: Statistics of synthetic benchmark.

Synthetic benchmark

We generate a synthetic benchmark in a manner similar to Santos (1994). We first generate a random Directed Acyclic Graph (DAG) as follows:

1. Choose $N \sim \mathcal{U}(min\_n, max\_n), E \sim \mathcal{U}(N, max\_e), T \sim \mathcal{U}(6, 12)$, where $\mathcal{U}(a, b)$ is a discrete uniform distribution ranging from a to b
2. Create a random directed graph G with N nodes and E edges, and let $G' = \{(u, v) \mid (u, v) \in G, u < v\}$

Note that G' is guaranteed to be a DAG. From G', we generate input observations O and background knowledge B, regarding nodes as propositional atoms and edges as axioms:

1. Let B be $\{\}$, and O be a set of randomly picked T terminal nodes from G'
2. For each node n in G', do:
   (a) $B \leftarrow B \cup \{n \leftarrow etc_{n,0}\}$
   (b) Choose $M \sim \mathcal{U}(1, 6), C \sim \mathcal{U}(0, 1)$
   (c) Let S be a set of predecessor nodes of n
   (d) Pop M nodes, denoted by $n_1, n_2, ..., n_M$, from S
   (e) If $C = 0$: $B \leftarrow B \cup \{n \leftarrow n_1 \wedge n_2 \wedge ... \wedge n_M \wedge etc_{n,1}\}$
   (f) If $C = 1$: $B \leftarrow B \cup \{n \leftarrow n_i \wedge etc_{n,i} \mid 1 \leq i \leq M\}$
   (g) Go to line (d) unless S is empty

We generated two types of benchmark problems: (i) SMALLBENCH: $min\_n = 50, max\_n = 100, max\_e = 100$ and (ii) LARGEBENCH: $min\_n = 1000, max\_n = 40000, max\_e = 600000$, which were motivated by previous applications of abductive reasoning (Ovchinnikova et al. 2011; Gordon 2016). The statistics of the generated benchmark problems are provided in Table 1.

We determined the prior probability of etcetera literals by using a Gaussian distribution with mean μ and variance σ. The assigned probabilities were made realistic by using different parameters according to the form of axiom: $\mu = 0.1, \sigma = 0.05$ for an axiom in which a solitary etcetera literal is its antecedent (e.g., $q \leftarrow etc_{q,0}$); otherwise $\mu = 0.5, \sigma = 0.05$ were used.

[1] `https://github.com/asgordon/EtcAbductionPy`

[2] `http://www.gurobi.com/`

System	Processing time (sec.)			Accuracy
	Avg.	Min.	Max.	
EtcAbductionPy	1.71	0.01	28.74	-
Proposed	**0.03**	0.01	0.08	100%

Table 2: Processing time and accuracy on SmallBench.

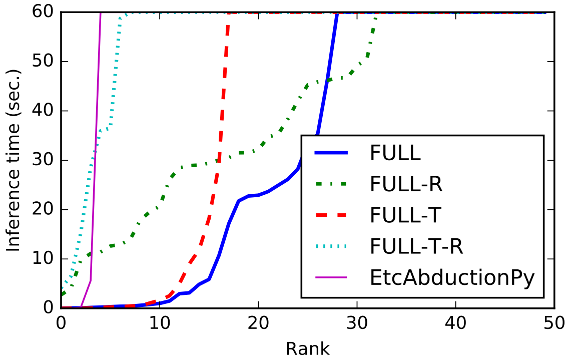

Figure 1: Rank-inference time distribution on LargeBench. R and T denote relevant reasoning and Tseytin transformation, respectively.

Results

We first compared the efficiency of both systems on SmallBench. For the proposed system, we also calculated the accuracy of the solutions by considering the outputs produced by EtcAbductionPy as a gold standard. The results are shown in Table 2. Both systems could find optimality-guaranteed solutions for all 50 problems on SmallBench. The results indicate that the proposed system significantly outperformed EtcAbductionPy and obtained exactly the same solutions as those of the reference implementation.

Next, we determined the scalability of the proposed approach by running both systems on LargeBench. For each problem, we plotted the inference time of three different instantiations of each system (Full: the proposed system, -T: "*without* Tseytin transformation," -R: "*without* relevant reasoning") and its rank in Figure 1. Our best system (Full-R) could find optimal solutions for 64.0 % (32/50) of problems within the given resource limit, whereas EtcAbductionPy achieved this for only 8.0 % (4/50) of problems. Moreover, the processing time of the proposed system is significantly shorter than that of EtcAbductionPy. The results indicate that our system is significantly more scalable than EtcAbductionPy.

In addition, we conducted an ablation study to determine the effect of relevant reasoning and the Tseytin transformation, the result of which is shown in Figure 1 (see Full-T, Full-R). We observe that both techniques strongly contribute to reducing the processing time, but relevant reasoning adversely affects the performance to some extent. This led us to speculate that the cost of relevant reasoning sometimes becomes much higher for large-scale problems.

Our system can be decomposed into four modules: (i) rel: relevant reasoning, (ii) com: Clark completion, (iii) I-gen: ILP translation, and (iv) I-opt: the optimization of Equation (16). Table 3 shows the average processing time of each module and the average size of generated ILP problems. The first row (Full) indicates that our bottle neck is on relevant reasoning and the generation of ILP problems.

System	rel/com	I-gen/I-opt	Total	Var.	Constr.
Full	3.4 / 1.1	4.4 / 3.0	12.0	53,397	76,716
Full	3.3 / **0.8**	**3.3 / 2.0**	9.4	39,500	56,426
Full-R	**0.4** / 2.7	14.4 / 6.4	23.8	176,629	251,892
Full	0.3 / **0.2**	**1.0 / 0.4**	2.0	13,183	18,953
Full-T	**0.2** / 1.9	2.1 / 0.5	4.7	22,716	25,887

Table 3: Average processing time of each module and average size of ILP problem.

Finally, we compared the average processing time and size of ILP problems of our systems (Full versus Full-R, Full versus Full-T). For fair comparison, we ran the systems on the same set of problems that were solved by both systems within the given resource limit (24, 17 problems, respectively). In this regard, for relevant reasoning (Full versus Full-R), we observe that the improvement is attributable to the Clark completion and ILP modules (com, I-gen, I-opt), which reduces the computational cost of relevant reasoning (rel). This indicates that identifying relevant axioms helps to reduce the complexity of an ILP problem. For the Tseytin transformation (Full versus Full-T), we observe that the improvement can mainly be attributed to the pre-processing modules (com, I-gen), but *not* to ILP optimization (I-opt). The Tseytin transformation did not help reduce the complexity of ILP problems in our benchmark, although it remains beneficial for the overall efficiency.

Related work

The SAT-based approach to abductive reasoning has not been widely explored in the literature. As described above, the translation of an abduction problem into deduction was reported previously (Console, Dupre, and Torasso 1991). In this paper, we present an extension of their result to translate an Etcetera Abduction problem into a Weighted-Max SAT problem, additionally representing the probabilistic evaluation function of Etcetera Abduction. Furthermore, this is the first time a large-scale evaluation of SAT-based abductive reasoning is conducted.

In the area of Artificial Intelligence, a number of prior studies were carried out to develop an efficient inference method for abductive reasoning (Santos 1994; Abdelbar and Hefny 2005; Inoue and Inui 2014, etc.). Unlike our work, however, these methods were tailored for abductive reasoning. Thus, finding a way to extend these frameworks to enhance their expressive inference capability (e.g., integrating forward reasoning) remains an open question. On the other hand, our framework is SAT-based, which allows us greater flexibility to extend the framework. For example, forward reasoning with uncertainty is naturally incorporated into the proposed framework by simply adding a weighted clause into ϕ. In addition, SAT-based formulation enables us to use highly advanced optimization algorithms developed in the field of SAT and Operations Research.

The work that most closely resembles our proposal is a series of studies that implemented abductive reasoning in Markov Logic Networks (Kate and Mooney 2009; Singla and Mooney 2011; Blythe et al. 2011), which is a popular well-developed probabilistic deduction framework (Richardson and Domingos 2006). Basically, these studies implemented abductive reasoning through the reverse implication of axioms in background knowledge, thereby sharing the same spirit as our Clark completion-based translation. The key difference is that they either do not model the evaluation function of Etcetera Abduction (Kate and Mooney 2009; Blythe et al. 2011), or they require a larger number of clauses to represent an abduction problem (e.g., the pairwise mutual exclusivity constraints in (Kate and Mooney 2009)). In addition, they do not evaluate their translation with a large-scale knowledge base.

Conclusions

We described Etcetera Abduction, a new implementation of logical abduction. We showed that an Etcetera Abduction problem can be translated into a Weighted Max-SAT problem, which enables us to leverage highly advanced technologies developed in the field of SAT and Operations Research, and to flexibly extend the framework. In our research, we used state-of-the-art mathematical optimizers developed in these fields and demonstrated the scalability of our proposal on a large-scale synthetic benchmark that contains up to ten thousand axioms. This is the first work to evaluate a SAT-based approach to abductive reasoning on such a large-scale knowledge base. The inference engine presented in this paper has been made publicly available at `https://github.com/naoya-i/EtcAbductionPy`. Our future work includes extending the proposed framework to accommodate first-order logic, and applying Etcetera Abduction to large-scale interpretation problems with application-specific knowledge bases.

Acknowledgement This work was supported by JSPS KAKENHI Grant Numbers 15H01702, 16H06614, CREST, JST and the Office of Naval Research, Grant Number N00014-16-1-2435.

References

Abdelbar, A. M., and Hefny, M. 2005. An efficient LP-based admissible heuristic for cost-based abduction. *Journal of Experimental and Theoretical Artificial Intelligence* 17(3):297–303.

Ansótegui, C., and Gabàs, J. 2013. Solving Weighted Partial MaxSAT with ILP. In *Proceedings of the 10th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 7874, 403–409.

Blythe, J.; Hobbs, J. R.; Domingos, P.; Kate, R. J.; and Mooney, R. J. 2011. Implementing Weighted Abduction in Markov Logic. In *Proceedings of the International Conference on Computational Semantics*, 55–64.

Charniak, E., and Goldman, R. 1991. A Probabilistic Model of Plan Recognition. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 160–165.

Clark, K. L. 1978. Negation as failure. In *Logic and data bases*. Springer. 293–322.

Console, L.; Dupre, D. T.; and Torasso, P. 1991. On the relationship between abduction and deduction. *Journal of Logic and Computation* 1(5):661–690.

Gordon, A. S. 2016. Commonsense Interpretation of Triangle Behavior. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 3719–3725.

Hobbs, J. R.; Stickel, M. E.; Appelt, D. E.; and Martin, P. 1993. Interpretation as abduction. *Artificial Intelligence* 63(1-2):69–142.

Inoue, N., and Inui, K. 2014. ILP-based Inference for Cost-based Abduction on First-order Predicate Logic. *Information and Media Technologies* 9(1):83–110.

Kate, R. J., and Mooney, R. J. 2009. Probabilistic Abduction using Markov Logic Networks. In *Proceedings of IJCAI Workshop on Plan, Activity and Intent Recognition*.

McCarthy, J. 1980. Circumscription - A form of non-monotonic reasoning. *Artificial Intelligence* 13(1):27–39.

Meadows, B.; Heald, R.; and Langley, P. 2015. An integrated account of explanation and question answering. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society*, 1571–1576.

Ovchinnikova, E.; Montazeri, N.; Alexandrov, T.; Hobbs, J. R.; Mccord, M. C.; and Mulkar-mehta, R. 2011. Abductive Reasoning with a Large Knowledge Base for Discourse Processing. In *Proceedings of the Ninth International Conference on Computational Semantics*, 225–234.

Ovchinnikova, E.; Gordon, A. S.; and Hobbs, J. R. 2013. Abduction for Discourse Interpretation: A Probabilistic Framework. In *Proceedings of the Joint Symposium on Semantic Processing*, 42–50.

Poole, D. 1993. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence* 64(1):81–129.

Richardson, M., and Domingos, P. 2006. Markov Logic Networks. *Machine learning* 62(1-2):107–136.

Santos, E. 1994. A linear constraint satisfaction approach to cost-based abduction. *Artificial Intelligence* 65(1):1–27.

Singla, P., and Mooney, R. J. 2011. Abductive Markov logic for plan recognition. In *Proceedings of the 25th National Conference on Artificial Intelligence*, 1069–1075.

Tseytin, G. S. 1983. On the complexity of derivation in propositional calculus. In *Automation of reasoning*. Springer. 466–483.

Yamamoto, K.; Inoue, N.; Inui, K.; Arase, Y.; and Tsujii, J. 2015. Boosting the Efficiency of First-Order Abductive Reasoning Using Pre-estimated Relatedness between Predicates. *International Journal of Machine Learning and Computing* 5(2):114–120.