Evaluating Abductive Hypotheses using an EM Algorithm on BDDs

Katsumi Inoue[1,2] **Taisuke Sato**[2,1] **Masakazu Ishihata**[2] **Yoshitaka Kameya**[2]
Hidetomo Nabeshima[3]

[1] Principles of Informatics Research Division, National Institute of Informatics, Japan
[2] Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Japan
[3] Division of Medicine and Engineering Science, University of Yamanashi, Japan

Abstract

Abductive inference is an important AI reasoning technique to find explanations of observations, and has recently been applied to scientific discovery. To find best hypotheses among many logically possible hypotheses, we need to evaluate hypotheses obtained from the process of hypothesis generation. We propose an abductive inference architecture combined with an EM algorithm working on binary decision diagrams (BDDs). This work opens a way of applying BDDs to compress multiple hypotheses and to select most probable ones from them. An implemented system has been applied to inference of inhibition in metabolic pathways in the domain of systems biology.

1 Introduction

Abductive inference is known as a reasoning method to generate best explanations of observations and is a pattern of reasoning that occurs in such diverse places as diagnosis, theory formation, language understanding and jury deliberation [Josephson and Josephson, 1994]. Recently, abductive reasoning has been well applied to scientific discovery in the area of *systems biology* [Zupan *et al.*, 2003; King *et al.*, 2004; Tran *et al.*, 2005; Tamaddoni-Nezhad *et al.*, 2006; Chen *et al.*, 2008]. In scientific domains, knowledge bases are often structured as a large *network*, in which relations among nodes have important meanings in applications. For example, in biological domains, a sequence of signalings or biochemical reactions constitutes a *pathway*, which specifies a mechanism to explain how genes or cells carry out their functions. Thagard [2003] describes how pathway-based explanations of diseases have frequently led to new treatments that diminish disease by enhancing or inhibiting pathways. Much information of pathways is available as public databases like KEGG [Kanehisa and Goto, 2000], but still knowledge bases are generally *incomplete* in these domains. Then, we need to predict the status of relations which is consistent with the status of nodes [Tamaddoni-Nezhad *et al.*, 2006], or augment unknown relations between nodes to explain observations [Zupan *et al.*, 2003; King *et al.*, 2004; Tran *et al.*, 2005]. These problems are characterized by abduction, and each set of inferred information to account for the observations is called an *explanation* or a *hypothesis*.

One salient feature in applications of abduction in systems biology is that, often, there are a large number of hypotheses that explain the observations. This situation occurs not only when the size of a network is huge, but there are two fundamental problems in these domains. First, knowledge is generally incomplete so that constraints among objects are not strong enough. For example, a possible status of each relation in a pathway is often chosen nondeterministically as a hypothesis. Considering all relations in a pathway as well as all pathways connecting to a goal node, there are a combinatorially large number of possible states that are consistent with the constraints. Second, multiple observations are often given at once rather than they are input sequentially. To solve such abductive tasks, a composite hypothesis must be assembled in such a way that one hypothesis is selected for each observation, which also results in an exponentially large number of composite hypotheses [Josephson and Josephson, 1994]. Hence, the use of abductive systems in such domains requires a method for *hypothesis evaluation* along with hypothesis generation. That is, we need to find *most plausible hypotheses* among the logically possible hypotheses.

Hence, in this work, we propose a new, simple yet powerful, computer-oriented model of abductive reasoning systems. This model consists of a *logically complete* hypothesis-generation system and a *statistical* hypothesis-evaluation system (Figure 1). The complete abductive procedure is realized with SOLAR [Nabeshima *et al.*, 2003], which is a sound and complete *consequence-finding procedure* for first-order full clausal theories. The statistical hypothesis evaluation is based on the *expectation-maximization* (EM) algorithm [Dempster *et al.*, 1977], which performs *maximum likelihood estimation* (MLE) on marginal distributions to estimate parameters (probabilities) in probabilistic models. As far as the authors know, no previous abductive reasoning system combined complete hypothesis generation with the EM algorithm.

After estimating atoms' probabilities by the EM algorithm, we use them to compute probabilities of competing hypotheses, and determine most probable hypotheses. However, we need to face the situation that the number of hypotheses computed by hypothesis generation is very large in general. To overcome this problem, we use an EM algorithm that works on *binary decision diagrams* (BDDs) [Ishihata *et al.*, 2008].

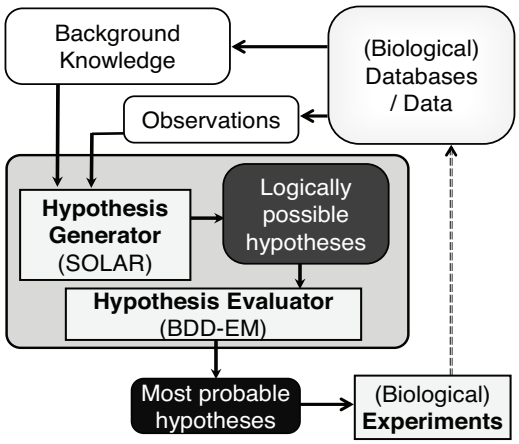

Figure 1: Abductive reasoning system

BDDs have been a basic tool for compactly representing boolean functions [Akers, 1978; Bryant, 1986]. Hence, our work opens a way of applying BDDs to compress multiple hypotheses and to select most probable ones from them.

An implemented abductive system has been applied to prediction of inhibitors in metabolic pathways in the domain of systems biology [Tamaddoni-Nezhad *et al.*, 2006]. The hypotheses that are ranked high by the proposed BDD-EM algorithm agree with experts' knowledge and known results, and contain those useful hypotheses that have been overlooked or even excluded by previous studies.

In the rest of this paper, Sections 2 and 3 respectively explain hypothesis generation and hypothesis evaluation in this work. Section 4 applies the proposed abductive system to reasoning about inhibition in metabolic networks. Sections 5 and 6 respectively discuss related and future work.

2 Hypothesis Generation

2.1 Logic of Explanation

The logical framework of hypothesis generation in abduction can be expressed as follows. Let B be a set of clauses, which represents the *background knowledge*, and O be a set of literals, which represents *observations* (or *goals*). Also let Γ be a set of literals representing the set of *abducibles*, which are candidate assumptions to be added to B for explaining O. Given B, O and Γ, the hypothesis-generation problem is to find a set H of literals (called a *hypothesis*) such that

$$B \cup H \models O, \tag{1}$$

$$B \cup H \text{ is consistent, and} \tag{2}$$

$$H \text{ is a set of instances of literals from } \Gamma. \tag{3}$$

In this case, H is also called an *explanation* of O (with respect to B and Γ). An explanation H of O is *minimal* if no proper subset of H satisfies the above three conditions. We often introduce additional conditions of hypotheses such as the maximum number of literals in each explanation. A hypothesis is *ground* if it is a set of ground literals.

The next proposition is important in our abductive framework: *if $H_1, \ldots, H_k$ are hypotheses that satisfy (1) and (2), then so does the disjunction of them, i.e.,*

$$B \cup \left\{ \bigvee_{i=1}^{k} H_i \right\} \models O, \text{ where } \bigvee_{i=1}^{k} H_i = \bigvee_{i=1}^{k} \bigwedge_{L \in H_i} L, \tag{4}$$

and $B \cup \{H_1 \vee \cdots \vee H_k\}$ is consistent. We call $H_1 \vee \cdots \vee H_k$ an *explanatory disjunction* for O, which can also be regarded as an explanation of O.

Hypothesis evaluation in Section 3 can be done only when hypotheses are ground. This restriction is often employed in applications whose observations are also given as ground literals. To guarantee that the number of minimal ground hypotheses is finite for any observation, we here assume that the language contains no function symbols and there are only finitely many constants.

2.2 Hypothesis Enumeration by SOLAR

Given the observations O, each explanation H of O can be computed by the principle of *inverse entailment* [Inoue, 1992; Muggleton, 1995], which converts the equation (1) to

$$B \cup \{\neg O\} \models \neg H, \tag{5}$$

where $\neg O = \bigvee_{L \in O} \neg L$ and $\neg H = \bigvee_{L \in H} \neg L$. Note that both $\neg O$ and $\neg H$ are clauses because O and H are sets of literals. Similarly, the equation (2) is equivalent to

$$B \not\models \neg H. \tag{6}$$

Hence, for any hypothesis H, its negated form $\neg H$ is deductively obtained as a "new" theorem of $B \cup \{\neg O\}$ which is not an "old" theorem of B alone. Moreover, by (3), every literal in $\neg H$ is an instance of a literal in $\overline{\Gamma} = \{\neg L \mid L \in \Gamma\}$.

SOLAR [Nabeshima *et al.*, 2003] is a sophisticated deductive reasoning system based on SOL-resolution [Inoue, 1992], which is complete for finding *minimal* consequences belonging to a given language bias (called a *production field*). Consequence-finding by SOLAR is performed by *skipping* literals belonging to a production field $\overline{\Gamma}$ instead of resolving them. Those skipped literals are then collected at the end of a proof, which constitute a clause as a logical consequence of the axiom set. Using SOLAR, we can implement an abductive system that is *complete* for finding minimal explanations due to completeness of consequence-finding. When a production field contains ground literals, they are converted to a non-ground production field by way of [Ray and Inoue, 2007] to assure completeness of ground hypotheses in abduction.

Although many other resolution-based abductive procedures are designed for Horn clauses or normal logic programs [Kakas *et al.*, 1998], SOLAR works for *full clausal theories* containing non-Horn clauses. Extending a *connection tableau* format [Letz *et al.*, 1994], SOLAR greatly avoides producing redundant deductions using various state-of-the-art pruning techniques [Nabeshima *et al.*, 2003], thereby enumeration of (negated) hypotheses is efficiently realized.

3 Hypothesis Evaluation

3.1 Basic Variables and the EM Algorithm

We here describe our hypothesis evaluation setting. Suppose that our problem is modeled with k independent boolean variables (or ground atoms) $X_1, X_2, \ldots, X_k$, each probabilistically taking either 1 (true) or 0 (false), and their boolean

function (or ground formula) $F = F(X_1, \ldots, X_k)$. We henceforth treat F as a boolean random variable and call it an *observable* (or *manifest*) *variable*. Contrastingly we call the X_i's *basic* (or *latent*) *variables*. We assume that only the value of F is observable while those of basic variables are not. The first step of hypothesis evaluation is to estimate *parameters*, i.e., probabilities of the basic variables from the manifestations of F using maximum likelihood estimation (MLE). However, as data is *incomplete*, meaning that we have no data about the basic variables as they are unobservable, a simple counting method for complete data does not work. Instead, we employ the expectation-maximization (EM) algorithm [Dempster *et al.*, 1977] which is applicable to the case of incomplete data like ours. The essence of the EM algorithm is to supplement missing data by their average. The output is parameters (locally) maximizing the likelihood of the manifestations. Since the EM algorithm is an abstract framework, we have to derive a concrete algorithm adapted to a specific probabilistic model. We next describe the EM algorithm adapted to our problem setting.

Let $\mathbf{X} = \{X_1, \ldots, X_k\}$ be the set of basic variables and $\phi(X) \in \{0, 1\}$ be the value of $X \in \mathbf{X}$ assigned by an assignment ϕ. We denote by Ψ the set of all assignments and by $\theta_{X=x}$ the parameter (probability) of X taking x ($\in \{0, 1\}$). $\vec{\theta}$ stands for the set of all parameters. Since the value $f \in \{0, 1\}$ of F is uniquely determined by an assignment $\phi \in \Psi$, F is considered as a function of ϕ. Hence, we can write the set of assignments for which F takes f as $F^{-1}(f) = \{\phi \in \Psi \mid F(\phi) = f\}$. We also introduce $\mathbf{1}_{\phi(X)=x}$ and let it take 1 if $\phi(X) = x$, and 0 otherwise.

Under this setting, the EM algorithm computes parameters $\vec{\theta}$ from $F^{-1}(f)$ and iterates their update until convergence. The update process consists of the *expectation step* (E-step) followed by the *maximization step* (M-step) as defined below.

E-step: Compute $E_{\vec{\theta}}[\mathbf{1}_{\phi(X)=x} | F = f]$, the conditional expectation for each $X \in \mathbf{X}$ and $x \in \{0, 1\}$, by

$$\frac{1}{P_{\vec{\theta}}(F = f)} \sum_{\phi \in F^{-1}(f)} \mathbf{1}_{\phi(X)=x} \prod_{X' \in \mathbf{X}} \theta_{X'=\phi(X')}$$

where $P_{\vec{\theta}}(F = f) = \sum_{\phi \in F^{-1}(f)} \prod_{X \in \mathbf{X}} \theta_{X=\phi(X)}$.

M-step:[1] Update $\theta_{X=x}$ for each $X \in \mathbf{X}$ and $x \in \{0, 1\}$ by

$$\hat{\theta}_{X=x} = \frac{E_{\vec{\theta}}[\mathbf{1}_{\phi(X)=x} | F = f]}{E_{\vec{\theta}}[\mathbf{1}_{\phi(X)=1} | F = f] + E_{\vec{\theta}}[\mathbf{1}_{\phi(X)=0} | F = f]} .$$

3.2 The BDD-EM Algorithm

The EM algorithm described in the previous subsection cannot deal with many variables because the E-step has to take the sum of $|F^{-1}(f)|$ terms, where $|F^{-1}(f)|$ is the number of assignments satisfying $F = f$, and $|F^{-1}(f)|$ is usually exponential in the number of variables. Often however, we can alleviate this intractability by expressing $F^{-1}(f)$ compactly using *binary decision diagrams* (BDDs) [Akers, 1978].

[1]The denominator at the M-step here is designed to work for a general case where iid variables occur more than once in the BDD.

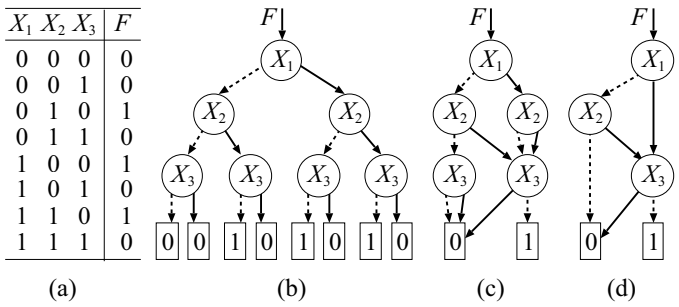

Figure 2: Examples of (a) a truth table, (b) a binary decision tree (BDT), (c) a BDD which is ordered but is not reduced, (d) the ROBDD, for $F = (X_1 \vee X_2) \wedge \neg X_3$.

A BDD is a rooted directed acyclic graph representing a boolean formula as a disjunction of exclusive conjunctions. It has two terminal nodes, $\boxed{1}$ (true) and $\boxed{0}$ (false). Each non-terminal node n is labeled with a propositional variable denoted by *Label*(n), and has two outgoing edges called 1-edge and 0-edge, indicating that *Label*(n) takes 1 and 0, respectively. To compactly express boolean formulas, we use a specific type of BDDs called *reduced ordered BDDs* (ROBDDs). When a variable ordering is fixed, ROBDDs give a unique representation of the target boolean formula [Bryant, 1986]. Figure 2 illustrates how the ROBDD of $F = (X_1 \vee X_2) \wedge \neg X_3$ is obtained. Figure 2 (a) is a truth table, in which a row corresponds to an assignment ϕ for $\mathbf{X} = \{X_1, X_2, X_3\}$. From the truth table, a *binary decision tree* (BDT) (Figure 2 (b)) is constructed, and then by applying two reduction rules, *deletion* and *merging*, repeatedly, we reach the ROBDD (Figure 2 (d)). ROBDDs give compressed representation of original boolean formulas because subexpressions are shared by other nodes and redundant nodes are deleted in the construction process. In what follows, BDDs mean ROBDDs.

Getting back to the EM algorithm, we introduce BDDs as a data structure to efficiently compute conditional expectations at the E-step. Since BDDs are directed and share subgraphs, sum-product computation of probabilities by dynamic programming becomes possible, just like the forward-backward probability computation used in hidden Markov models. The resulting EM algorithm is called the *BDD-EM algorithm* [Ishihata *et al.*, 2008]. It can compute $E_{\vec{\theta}}[\mathbf{1}_{\phi(X)=x} | F = f]$ in time proportional to the size of a BDD representing F at the E-step.

3.3 Computing Probabilities of Hypotheses

We now show how to evaluate hypotheses output by SOLAR using the BDD-EM algorithm. Given the background knowledge B and the observations O, suppose that SOLAR outputs the minimal ground explanations of O as $H_1, \ldots, H_k$. Our task is to rank the hypotheses $H_1, \ldots, H_k$ according to their probabilities. So we need to compute their probabilities.

Let $\mathcal{A}$ be a finite set of ground atoms $\{A_1, \ldots, A_n\}$ such that the set of abducibles Γ satisfies that $\Gamma \subseteq \mathcal{A} \cup \{\neg A \mid A \in \mathcal{A}\}$. We consider each $A_i \in \mathcal{A}$ ($i = 1, \ldots, n$) as a basic boolean variable, and put $\theta_i = \theta_{A_i=1} = P(A_i)$. The θ_i's determine the probability of every boolean formula made

up of $\{A_1, \ldots, A_n\}$ including $H_1, \ldots, H_k$ and we are free to choose their values. At this point, recall that we wish to obtain plausible hypotheses that explain our observations. In this context, false hypotheses are useless and the parameter $\theta_1, \ldots, \theta_n$ should be chosen so that the probability of one of $H_1, \ldots, H_k$ being true is maximum. In other words, the explanatory disjunction $H_1 \vee \cdots \vee H_n$ for O in (4) should have a high probability. Furthermore, since the explanatory disjunction cannot prove O without the help of the background knowledge B, B also should have a high probability. Hence, we will search for the θ_i's that maximize the probability of $(H_1 \vee \cdots \vee H_n) \wedge B$. This is done by applying the EM algorithm to

$$F = \left(\bigvee_{i=1}^{k} H_i\right) \wedge \left(\bigwedge_{C \in Ground(B)} C\right) \tag{7}$$

as the observable boolean function, where $Ground(B)$ is the set of all ground instances of clauses in B. In fact, the equation (4) implies that $P(F) \leq P(O)$, then by maximizing a lower bound $P(F)$, $P(O)$ is expected to be maximized too.

Unfortunately, the ground formula F in (7) is quite large. We therefore employ the BDD-EM algorithm which retains F as a BDD, but in some cases the size of $Ground(B)$ is still too large to store in the BDD. Then we introduce a "proof-theoretic" approximation of $Ground(B)$ to reduce its size. For each explanation H_i $(1 \leq i \leq k)$ of O, let B_i be a set of ground clauses in $Ground(B)$ that contribute to an abductive proof of O with H_i: $B_i \cup H_i \vdash O$. In SOLAR, this proof can be extracted as a ground deduction of $\neg H_i$ from $B \cup \{\neg O\}$. Then,

$$F' = \left(\bigvee_{i=1}^{k} H_i\right) \wedge \left(\bigwedge_{i=1}^{k} \bigwedge_{C \in B_i} C\right) \tag{8}$$

is substituted for (7) as an approximation of F. Here, $B_1 \cup \cdots \cup B_k$ is reasonably expected as a good approximation of B as far as O and $H_1, \ldots, H_k$ are concerned. In fact, if a ground instance of a clause in B does not appear in any abductive proof of O, it is irrelevant to our abductive task, thereby can be excluded from parameter learning in the EM algorithm.

Equating $\mathcal{A}$ with the set of atoms appearing in F', the BDD-EM algorithm estimates the probabilities of ground atoms in $\mathcal{A}$ as maximizers of the probability of F'. The individual probability of H_i $(i = 1, \ldots, k)$ is then computed as the product of the probabilities of literals appearing in H_i, which is used to rank the H_i's.

4 Evaluation: Reasoning about Inhibition in Metabolic Networks

A metabolic pathway is a coherent sequence of enzymatic reactions which are interconnected via substrates. To represent inhibitory effects on metabolic pathways, a logical model has been introduced in [Tamaddoni-Nezhad *et al.*, 2006]. This model has been used in the Metalog project for predictive toxicology [Tamaddoni-Nezhad *et al.*, 2006], in which actions of toxins are predicted from NMR together with known pathway information in KEGG [Kanehisa and Goto, 2000]. In the logical model, three kinds of biochemical information are represented in a clausal form: (a) inhibitory effects of reactions in terms of qualitative concentration changes, (b) integrity constraints on inhibitory effects and concentration changes, and (c) chemical reactions catalyzed by enzymes in metabolic networks. Part of the data set was available from the web site containing (a) 4 rules of causal effects, (b) 4 integrity constraints, and (c) 76 facts of enzymatic reactions which are all reversible. These data are represented as Horn clauses in the background knowledge. The next clauses are examples of the 4 causal rules, 2 integrity constrains, and 4 reactions:

$reaction(X, Enz, Y) \wedge inhibited(Enz, \mathtt{t}, Y, X, T)$
$\quad \rightarrow concentration(X, down, T).$
$reaction(X, Enz, Y) \wedge inhibited(Enz, \mathtt{f}, Y, X, T) \wedge$
$\quad concentration(Y, down, T) \rightarrow concentration(X, down, T).$
$reaction(X, Enz, Y) \wedge inhibited(Enz, \mathtt{t}, Y, X, T)$
$\quad \rightarrow concentration(Y, up, T).$
$reaction(X, Enz, Y) \wedge inhibited(Enz, \mathtt{f}, Y, X, T) \wedge$
$\quad concentration(Y, up, T) \rightarrow concentration(X, up, T).$
$\neg concentration(M, up, T) \vee \neg concentration(M, down, T).$
$\neg inhibited(Enz, \mathtt{t}, X, Y, T) \vee \neg inhibited(Enz, \mathtt{f}, X, Y, T).$
$reaction(2\text{-}oxo\text{-}glutarate, 2.6.1.39, l\text{-}2\text{-}aminoadipate).$
$reaction(l\text{-}2\text{-}aminoadipate, 2.6.1.39, 2\text{-}oxo\text{-}glutarate).$
$reaction(2\text{-}oxo\text{-}glutarate, 1.1.1.42, isocitrate).$
$reaction(l\text{-}as, 4.3.2.1, arginine).$

The data set contains 100 observations for this network including the following concentration changes:

$concentration(citrate, down, 8).$
$concentration(2\text{-}oxo\text{-}glutarate, down, 8).$
$concentration(l\text{-}2\text{-}aminoadipate, up, 8).$
$concentration(l\text{-}as, up, 8).$

The abducibles in this example is given as the schema $\{inhibited(Enzyme, Status, From, To, Time)\}$, any ground instance of which can be assumed. In [Tamaddoni-Nezhad *et al.*, 2006], Progol [Muggleton, 1995] (ver.5.0) is used for computing abductive hypotheses.

With this data set, SOLAR computes 808 abductive derivations in the form (5) for the observations with respect to $Time = 8$ (hrs), of which 66 hypotheses are consistent. The maximum search depth is set to 5 and the maximum length of produced clauses is set to 15 in running SOLAR. In the 66 hypotheses, the unique output of Progol5.0 is contained.

Next, the BDD-EM algorithm takes as the input F' the disjunctive normal form of 66 hypotheses obtained by SOLAR and the ground instances used in abductive proofs of the observations. In this process, to satisfy the integrity constraint that two ground atoms $inhibited(e, \mathtt{t}, m1, m2, 8)$ and $inhibited(e, \mathtt{f}, m1, m2, 8)$ cannot occur simultaneously, we convert them to the complimentary literals $inhibited(e, m1, m2, 8)$ and $\neg inhibited(e, m1, m2, 8)$, respectively. There are 70 variables (ground atoms $\mathcal{A}$) for this input, and the ROBDD contains 384 nodes. After repeatedly running the EM algorithm on the BDD 100,000 times with random initialization, the probabilities of abducibles were chosen as the one achieving the highest likelihood.[2]

[2]We excluded the top two extremely biased ones that only have less than 10 hypotheses with probabilities more than $1\mathrm{E}{-10}$.

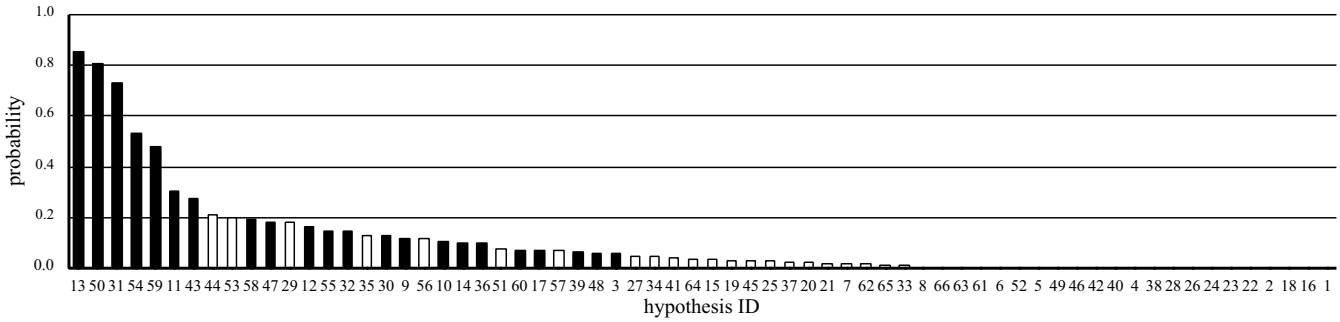

Figure 3: Ranking of 66 hypotheses

The ranking of 66 hypotheses in order of probability is plotted in Figure 3.[3] In the figure, the hypotheses are indexed as 1, 2, . . ., 66, in the order of appearance in the output of SOLAR. In our result, the hypothesis #13 is put in the first rank with the probability 0.85. This best one agrees with both experts' knowledge and a known result written in [Tamaddoni-Nezhad *et al.*, 2006], that is,

1. The enzyme EC2.6.1.39 is inhibited by hydrazine, and
2. The enzyme EC4.3.2.1 is inhibited by hydrazine,

which are respectively represented as the abducibles:

$$inhibited(2.6.1.39, \mathtt{t}, l\text{-}2\text{-}aminoadipate, 2\text{-}oxo\text{-}glutarate, 8).$$
$$inhibited(4.3.2.1, \mathtt{t}, l\text{-}as, fumarate, 8).$$

Moreover, the average probability of the 22 hypotheses satisfying these two inhibitory states (shown by black bars) is 0.258, which is 8.19 times higher than the average probability 0.032 of the 44 hypotheses not satisfying both of them (shown by white bars). More impressively, the top 7 hypotheses in the ranking satisfy these two inhibitory states. Hence, we can see that those biologically reasonable explanations are more probable than others in this logical model.

On the other hand, Progol attempts to find the most compressive hypotheses based on its internal predictive measure, so avoids producing all consistent hypotheses. In fact, when Progol is run without any option, it outputs the unique hypothesis #12, which now ranks 13th in our experiment. This pruning mechanism by Progol gains efficiency, but is in danger of losing many alternative hypotheses which are competing with the best ones. The results here indicate that our method of hypothesis evaluation provides a more precise way of hypothesis ranking,[4] and thus can provide the user better promising hypotheses and insights into the next experiment.

5 Related Work and Discussion

Previous work on combining abduction and the EM algorithm can be seen in PRISM [Sato and Kameya, 2001], which realizes *statistical abduction* from probabilistic logic programs. PRISM uses definite clauses similarly to Probabilistic Horn abduction in [Poole, 1993] and realizes efficient proof search through a tabling mechanism, though not applicable to non-Horn clausal theories. PRISM employs a propositionalized data structure called *explanation graphs* to represent propositional formulas in DNF. However, PRISM assumes the *exclusiveness condition* that the disjuncts are exclusive to make sum-product probability computation possible. In our work, on the other hand, SOLAR can deal with abduction in non-Horn clausal theories, and the BDD-EM algorithm does not assume the exclusiveness condition. ProbLog [De Raedt *et al.*, 2007] is a recent probabilistic logic programming language that computes probabilities via BDDs. A ProbLog program computes the probability of a query atom by applying sum-product computation to a BDD which represents independent choices of program clauses. ProbLog also allows definite clauses only, and is not designed for abduction.

For abduction in propositional theories, Simon and del Val [2001] propose a consequence-finding procedure implemented on Zero-suppressed BDDs. Eiter and Makino [2002] present an efficient algorithm for hypothesis enumeration in propositional Horn theories. Since our BDD-EM algorithm works on ground theories, it is surely possible to give outputs of these algorithms to the input of the BDD-EM algorithm for evaluating them. The problem here would be how to reduce the size of ground instances of the background theory when it is given as a first-order theory like the pathway representation in Section 4. Our proof-theoretic approximation F' by (8) gives a compromise to this matter, although it seems hard to extract abductive proofs from the algorithms in [Simon and del Val, 2001; Eiter and Makino, 2002].

The EM algorithm has been applied to constraint satisfaction problems (CSPs) in [Hsu *et al.*, 2007]. CSPs and abduction are common in finding assignments of variables: in abduction we find probable assignments of abducibles, while likely variable assignments are estimated in CSPs. However, the goal of [Hsu *et al.*, 2007] is to find a solution of a difficult CSP, which is different from our evaluation of explanations that have been already computed by hypothesis generation.

Finally, abduction has been applied to various problems in systems biology [Zupan *et al.*, 2003; King *et al.*, 2004; Tran *et al.*, 2005; Tamaddoni-Nezhad *et al.*, 2006; Chen *et al.*, 2008]. However, none of these work has employed a complete hypothesis generator followed by a systematic hy-

[3] In the submitted version, the program used in the experiment had a bug while a similar tendency was observed. The result in Figure 3 obtained with a corrected program is better than the previous.

[4] We have also obtained the resluts for $Time = 24, 48, 72, 96$, in which 1638, 3738, 22 and 5145 hypotheses are respectively ordered.

pothesis evaluator in contrast to our use of the EM algorithm. Those underlying abductive systems do hypothesis generation by evaluating hypotheses at the same time. Although this reduces the search space, incomplete search may overlook useful hypotheses. In these domains, the process of hypothesis evaluation should be more clearly shown to the user. We believe that our abductive architecture provides a new way of hypothesis generation and evaluation for this purpose.

6 Conclusion

We have proposed a novel abductive reasoning architecture which has several unique features. SOLAR provides a complete system for hypothesis generation from full clausal theories, and the BDD-EM algorithm provides a systematic and domain-independent way of hypothesis evaluation in which multiple ground hypotheses are compressed in a BDD. No previous abductive system contains these features, and the necessity of these features are justified in the domain of systems biology. We are applying the proposed abductive system to other important problems in systems biology.

As future work, we plan to use some propositional algorithms for hypothesis generation, and to apply the BDD-EM algorithm for hypothesis evaluation in statistical abduction [Poole, 1993; Sato and Kameya, 2001; Chen *et al.*, 2008].

Acknowledgments

This research is supported in part by the 2007–2009 JST-CNRS Strategic Japanese-French Cooperative Program and by the 2008-2011 JSPS Grant-in-Aid for Scientific Research (A) No. 20240016. We thank Andrei Doncescu, Koji Iwanuma, Stephen Muggleton, Oliver Ray, Takehide Soh and Yoshitaka Yamamoto for many discussions and help.

References

[Akers, 1978] Akers, S.B., Binary decision diagrams, *IEEE Trans. Computers*, 27(6):509–516, 1978.

[Bryant, 1986] Bryant, R., Graph-based algorithms for boolean function manipulation, *IEEE Trans. Computers*, 35(8):677–691, 1986.

[Chen *et al.*, 2008] Chen, J., Muggleton, S. and Santos, J., Learning probabilistic logic models from probabilistic examples, *Machine Learning*, 73:55–85, 2008.

[Dempster *et al.*, 1977] Dempster, A., Laird, N. and Rubin, D., Maximum likelihood from incomplete data via the EM algorithm, *J. Royal Statistical Society*, B 39:1–38, 1977.

[De Raedt *et al.*, 2007] De Raedt, L., Angelika, K. and Toivonen, H., ProbLog: a probabilistic Prolog and its application in link discovery, in: *Proc. of IJCAI-07*, pp.2468–2473, 2007.

[Eiter and Makino, 2002] Eiter, T. and Makino, K., On computing all abductive explanations, in: *Proc. of AAAI-02*, pp.62–73, 2002.

[Hsu *et al.*, 2007] Hsu, E., Kitching, M., Bacchus, F. and McIlraith, S., Expectation maximization to find likely assignments for solving CSP's, in: *Proc. of AAAI-07*, pp.224–230, 2007.

[Inoue, 1992] Inoue, K., Linear resolution for consequence finding, *Artificial Intelligence*, 56(2,3):301–353, 1992.

[Ishihata *et al.*, 2008] Ishihata, M., Kameya, Y., Sato, T. and Minato, S., Propositionalizing the EM algorithm by BDDs, in: *Late Breaking Papers of the 18th Int'l Conf. on Inductive Logic Programming*, pp.44–49, 2008.

[Josephson and Josephson, 1994] Josephson, J.J. and Josephson, S.G., *Abductive Inference: Computation, Philosophy, Technology*, Cambridge Univ. Press, 1994.

[Kakas *et al.*, 1998] Kakas, A.C., Kowalski, R.A. and Toni, F., The role of abduction in logic programming, in: *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol.5, pp.235–324, Oxford Univ. Press, 1998.

[Kanehisa and Goto, 2000] Kanehisa, M. and Goto, S., KEGG: Kyoto encyclopedia of genes and genomes, *Nucleic Acids Research*, 28:27–30, 2000.

[King *et al.*, 2004] King, R.D., Whelan, K.E., Jones, F.M., Reiser, P.G.K., Bryant, C.H., Muggleton, S.H., Kell, D.B. and Oliver, S.G., Functional genomic hypothesis generation and experimentation by a robot scientist, *Nature*, 427:247–252, 2004.

[Letz *et al.*, 1994] Letz, R., Goller, C. and Mayr, K., Controlled integration of the cut rule into connection tableau calculi, *J. Automated Reasoning*, 13:297–338, 1994.

[Muggleton, 1995] Muggleton, S., Inverse entailment and Progol, *New Generation Computing*, 13:245–286, 1995.

[Nabeshima *et al.*, 2003] Nabeshima, H., Iwanuma, K. and Inoue, K., SOLAR: a consequence finding system for advanced reasoning, in: *Proc. of TABLEAUX '03*, LNAI, 2796, pp.257–263, Springer, 2003.

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

[Ray and Inoue, 2007] Ray, O. and Inoue, K., A consequence finding approach for full clausal abduction, in: *Proc. of the 10th Int'l Conf. on Discovery Science*, LNAI, 4755, pp.173–184, Springer, 2007.

[Sato and Kameya, 2001] Sato, T. and Kameya, Y., Parameter learning of logic programs for symbolic-statistical modeling, *J. Artif. Intell. Res.*, 15:391–454, 2001.

[Simon and del Val, 2001] Simon, L. and del Val, A., Efficient consequence finding, in: *Proc. of IJCAI-01*, pp.359–365, 2001.

[Thagard, 2003] Thagard, P., Pathways to biomedical discovery, *Philosophy of Science*, 70:235-254, 2003.

[Tamaddoni-Nezhad *et al.*, 2006] Tamaddoni-Nezhad, A., Chaleil, R., Kakas, A. and Muggleton, S., Application of abductive ILP to learning metabolic network inhibition from temporal data, *Machine Learning*, 65:209–230, 2006.

[Tran *et al.*, 2005] Tran, N., Baral, C., Nagaraj, V.J. and Joshi, L., Knowledge-based framework for hypothesis formation in biochemical networks, *Bioinformatics*, 21(Suppl.2):ii213–ii219, 2005.

[Zupan *et al.*, 2003] Zupan, B., Demšar, J., Bratko, I., Juvan, P., Halter, J., Kuspa, A. and Shaulsky, G., GenePath: a system for automated construction of genetic networks from mutant data, *Bioinformatics*, 19(3):383–389, 2003.