On Minimality and Integrity Constraints in Probabilistic Abduction

Calin-Rares Turliuc, Nataly Maimari, Alessandra Russo, and Krysia Broda

Department of Computing, Imperial College London, United Kingdom
{ct1810,nataly.maimari08,a.russo,k.broda}@imperial.ac.uk

Abstract. Abduction is a type of logical inference that can be successfully combined with probabilistic reasoning. However, the role of integrity constraints has not received much attention when performing logical-probabilistic inference. The contribution of our paper is a probabilistic abductive framework based on the distribution semantics for normal logic programs that handles negation as failure and integrity constraints in the form of denials. Integrity constraints are treated as evidence from the perspective of probabilistic inference. An implementation is provided that computes alternative (non-minimal) abductive solutions, using an appropriately modified abductive system, and generates the probability of a query, for given solutions. An example application of the framework is given, where gene network topologies are abduced according to biological expert knowledge, to probabilistically explain observed gene expressions. The example shows the practical utility of the proposed framework.

Keywords: abductive logic programming, probabilistic abduction, distribution semantics.

1 Introduction

Abductive reasoning is a method of logical inference which explains observations (or queries) by making assumptions on possible facts, called abducible atoms. Abduction has been used in various applications [13], e.g. diagnosis, high-level vision, natural language understanding, planning, knowledge assimilation, etc. The choice of the assumptions is often filtered through integrity constraints, i.e. rules which eliminate certain solutions. A solution of an abductive task is therefore a set of abducible atoms that do not violate the integrity constraints and that if true make the query valid. Abductive solutions are hypotheses and as such are inherently uncertain. For a given abductive task there may be multiple solutions which may be ranked according to some notion of plausibility.

In a model governed by uncertainty, it is reasonable to consider a probability distribution over the truth values of each (ground) abducible. This probabilistic perspective provides a method of quantitatively estimating the quality of the abductive solutions, and, consequently, that of the solved query. Introducing probability in abduction essentially redefines the notion of abductive solutions as no longer the minimal but the *most preferred* (possibly non minimal) assumptions, based on their probability, needed to explain a given query. Most existing work in probabilistic abduction (cf. Section 5) does not discuss minimality, nor the role of integrity constraints. In this paper, we treat integrity constraints as evidence from the perspective of probabilistic inference (i.e. the goal is to compute $P(Q|E)$, where Q is the query, and E the evidence). Typically, the query is a set of random variables, and the evidence is a set of random variables whose outcome is observed. We extend this notion of evidence to a set of constraints imposed on the model, expressed as a logical formula. To motivate the main features of our probabilistic abductive approach, including dropping the minimality requirement, consider the following example of an abductive task:

Example 1. In the rules below the abducible atoms are rained_last_night and sprinkler_was_on.

$$\begin{aligned} \text{grass\_is\_wet} &\leftarrow \text{rained\_last\_night} \\ \text{grass\_is\_wet} &\leftarrow \text{sprinkler\_was\_on} \\ \text{shoes\_are\_wet} &\leftarrow \text{grass\_is\_wet} \end{aligned}$$

The explanations of the observation that the shoes_are_wet are that either it rained_last_night or the sprinkler_was_on.

In the above example, the explanation that it rained_last_night *and* the sprinkler_was_on is non-minimal. We argue that if abduction is augmented with probability using the distribution semantics, non-minimal solutions contribute to the probability of the query, and thus cannot be discarded. Suppose that we know that there is a probability 0.6 that it rained_last_night (with the complementary probability if the abducible is false), and 0.7 that the sprinkler_was_on (with the same remark). One might be tempted to choose the latter explanation, based on its higher probability. However, if rained_last_night and sprinkler_was_on are independent random events, the joint probability of rained_last_night and sprinkler_was_on is computed as shown in Table 1. Under this assumption, the most probable scenario is that it rained last night *and* the sprinkler was on. So the explanation with highest probability is not necessarily the minimal one.

Table 1. Joint probability on the abducibles in Example 1

rained_last_night	sprinkler_was_on	P(rained_last_night, sprinkler_was_on)
false	false	0.12
false	true	0.28
true	false	0.18
true	true	0.42

Furthermore, most semantics for abduction would interpret the explanation rained_last_night as rained_last_night is true and sprinkler_was_on is false, and similarly for the explanation sprinkler_was_on, i.e. all the abducibles in the explanation are true, and all that are not in it are false. The probability of the observation shoes_are_wet is 0.88 (i.e. sum of the join probabilities in the last three rows) as in the last three interpretations in Table 1 shoes_are_wet is true. Computing this probability means, therefore, asking the probability that shoes_are_wet is true under any explanation. This implies that choosing one explanation over another is no longer arbitrary, or according to minimality or other criteria (e.g. Example 2.1 in [13]). Instead, each explanation contributes with a probability mass towards the probability of the observation and all possible explanations should be considered. If a choice of particular explanations is required, then the one with the highest probability should be preferred.

Example 1 shows also that to compute the correct probability of a given observation or query, the closed world assumption (CWA) on abducibles is insufficient. In Table 1, the last interpretation would not be covered by the CWA over minimal explanations. In our approach, we propose an open world interpretation of abducibles (cf. Section 3).

Let us now assume Example 1 to be extended with the integrity constraint $\leftarrow$ sprinkler_was_on., expressing the statement that the sprinkler was off. This implies that the only explanation will be rained_last_night. Treating integrity constraints as evidence means computing the probabilistic inference $P(Q|IC)$. In our example $Q = \{\text{shoes\_are\_wet}\}$ and the conditional probability $\frac{P(Q,IC)}{P(IC)}$ is, in this case, given by $\frac{0.18}{0.18+0.12} = 0.6$, which is indeed the expected result of the probability of rained_last_night. If we, instead, extend Example 1 with the integrity constraint $\leftarrow$ not rained_last_night., meaning that we know that it rained last night, then the probability of shoes_are_wet is $\frac{0.18+0.42}{0.18+0.42} = 1$. In summary, the contributions of this papers are:

1. a probabilistic abductive framework, based on the distribution semantics for normal logic programs [20,22], that handles negation as failure and integrity constraints in the form of denials, and provides an open world interpretation of abducibles;
2. a procedure for logical-probabilistic inference, based on the ASystem [14,17];
3. a practical application in the context of gene networks.

The paper is organized as follows. Section 2 introduces our framework and define our probability model by adapting the distribution semantics for normal logic programs under Fitting semantics [22]. In Section 3 we provide an implementation of our framework that uses an existing state-of-the-art abductive system, appropriately modified in order to support the computation of non-minimal abductive solutions. Section 4 illustrates the applicability of our framework to the real world problem of gene network inference from observed data. Networks are abduced as directed graphs with probabilistic edges to explain observed gene expressions. We learn the probabilities of the edges (gene interactions) that would maximize the probability of a given query and interpret the results. Section 5 discusses related work. In Section 6 we present future work and conclude.

2 Distribution Semantics for an Abductive Framework

An *abductive framework* is a tuple $\langle P, AB, IC \rangle$, where P is a normal logic program, AB is a possibly infinite set of ground atoms called *abducibles*, and IC is a set of integrity constraints expressed as denials, each having the form $\forall \overline{X} \leftarrow \Gamma.$, where Γ is a set of literals and $\overline{X}$ is a set of variables. A query Q is a conjunction of existentially quantified literals and denials. An abductive solution for a query Q is a set of abducibles Δ, such that the ground instantiations of Δ, denoted $ground(\Delta)$ are elements in AB and:

- $comp_3(P \cup \Delta) \models Q$.
- $comp_3(P \cup \Delta) \models IC$.
- $comp_3(P \cup \Delta) \models CET$

where CET denotes the Clark Equality Theory axioms [2], and $comp_3(\Pi)$ the Fitting three-valued completion of a program Π [9].

We define our probabilistic abductive framework by integrating distribution semantics [22] into the above notion of an abductive framework. Informally, distribution semantics defines a probability distribution over the set of interpretations over a set of facts F and extends it to a probability distribution over interpretations of a program Π by applying the Fitting fixpoint operator [9]. This extension implies that the probability of an interpretation of the facts $I \subseteq F$ will have the same value as the probability of an interpretation I_Π of Π, given that I_Π is the fixpoint of I according to the rules in Π. In a similar fashion, we consider a two-valued interpretation over abducibles $I \subseteq AB$ and extend it to interpretations I_Π of the Herbrand base of the whole program Π. The interpretation I_Π is in general three-valued, however we impose the restriction that I_Π is two-valued, and in what follows we will treat it as such. We then consider a probability distribution P_{AB} with the sample space the set of all ground interpretations of abducibles (i.e. the powerset of AB) and we extend P_{AB} to a probability distribution P_Π with the sample space the set of all the ground interpretations of the Herbrand base of Π. To compute P_{AB}, we assume that the assignments of truth values to an abducible are independent events, and that all abducibles are *basic*, i.e. they do not appear in the heads of the rules in Π [13]. If each abducible $\delta \in AB$ has a probability $P(\delta)$ of being true (and a probability $1 - P(\delta)$ of being false), then P_{AB} is computed as:

$$P_{AB}(I) = \prod_{\delta \in I} P(\delta) \prod_{\delta \notin I} (1 - P(\delta))$$

P_{AB} is then extended to a probability distribution P_Π by applying Fitting's fixpoint operator Φ_Π to reach the fixpoint Φ_Π^∞ [9].

$$P_\Pi(I_\Pi) = \begin{cases} P_{AB}(I) & \text{if } I_\Pi = \Phi_\Pi^\infty(I) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Given the above probability distribution, it is possible to compute the probability of a two-valued interpretation I_B of a set B of ground atoms in Π by marginalization:

$$P_\Pi(I_B) = \sum_{I_\Pi \text{ s.t. } I_B \subseteq I_\Pi} P_\Pi(I_\Pi)$$

For a single atom a, we write $P_\Pi(a)$ with the meaning $P_\Pi(\{a\})$ and $P_\Pi(\neg a)$ with the meaning $P_\Pi(\emptyset)$.

In Example 1, P_{AB} is the last column of Table 1, and the sample space is given by the other columns. P_Π is obtained by extending P_{AB} over grass_is_wet and shoes_are_wet, with the appropriate truth values, i.e. the or function of rained_last_night and sprinkler_was_on. For all other interpretations, P_Π is 0 (Equation 1).

The probability of a query Q given evidence expressed as integrity constraints IC is then:

$$P_\Pi(Q|IC) = \frac{P_\Pi(Q, IC)}{P_\Pi(IC)} \tag{2}$$

$$P_\Pi(Q, IC) = \sum_{\substack{I_\Pi \text{ s.t. } Q \subseteq I_\Pi \\ I_\Pi \models IC}} P_\Pi(I_\Pi) \tag{3}$$

$$P_\Pi(IC) = \sum_{I_\Pi \models IC} P_\Pi(I_\Pi) \tag{4}$$

Informally, $P_\Pi(Q|IC)$ is the ratio of the probability of the interpretations that agree with Q and do not violate the integrity constraints (Equation 3) over the probability of the interpretations that do not violate the integrity constraints (Equation 4).

The novel aspect of our approach is the definition of evidence as a set of integrity constraints, inspired by Markov Logic Networks [7] where the notions of query and evidence are generalized to first-order formulae. This is more expressive than traditional definitions of evidence (i.e. conjunction of random variables taking particular values), because denials can express statements like "random variables X and Y cannot take values x, respectively y at the same time".

In the following section, we describe a logical-probabilistic procedure based on the ASystem [14,17] which can be used for the inference of $P_\Pi(Q|IC)$.

3 A Probabilistic Abductive System

This section describes the implementation of our probabilistic abductive framework. It builds upon an existing abductive system, called ASystem [14,17], briefly described in Section 3.1, and adapts it in Section 3.2 to allow non minimal abductive solutions. Section 3.3 shows how the abductive answers are used for probabilistic inference.

3.1 A Brief Description of the ASystem

The proof procedure of the ASystem [14,17] can be viewed as a *state rewriting* process, where each state rewrite is driven by the application of *inference rules*. The latter handle also *finite domain and real constraints* using a constraint solver. The system can compute *non-ground answers* and uses *constructive negation*, instead of standard negation as failure. Its development was inspired by other abductive systems such as SLDNFA [6], ACLP [12], IFF [10]. The semantics used in the ASystem is the three-valued completion semantics ($comp_3$) [9]: an interpretation of the abducibles is implicitly two-valued, whereas the interpretation of the predicates in P is three-valued. The proof procedure can be viewed as a tree, where the nodes are *ASystem states* and each node generates children nodes according to a set of *inference rules* and a *selection strategy*. The root of the tree is the *initial state*, and the leaves are *failure states* or *success states*.

Definition 1 (ASystem state). *An ASystem state S is a tuple $(\mathcal{G}, \mathcal{ST})$.*

- *$\mathcal{G}$ is a set of goals where each goal can be a literal or a denial. All the variables except the ones universally quantified in the denials are existentially quantified.*
- *$\mathcal{ST}$ is a tuple $(\Delta, \mathcal{N}, \mathcal{E}, \mathcal{C})$ of four stores: Δ is the abducible store, a set of (non-ground) abducible atoms, $\mathcal{N}$ is the denial store, a set of denials (or dynamic integrity constraints), $\mathcal{E}$ is a set of (in)equalities, $\mathcal{C}$ is a set of finite domain or real constraints.*

A *selection strategy* Ξ has a two-fold role: it selects a goal G_i from the set $\mathcal{G}$, and if the goal is a denial $\forall\overline{Y} \leftarrow \Gamma$. it further selects a literal from Γ. A selection strategy is called *safe* if, in a failure goal, it never selects a negative literal or a constraint literal, if the arguments of the predicate include a universally quantified variable. If a failure goal contains only universally quantified negative literals and universally quantified constraint literals, the derivation using a safe selection strategy *flounders* and fails.

Definition 2 (Meaning of an ASystem state). *The meaning of an ASystem state $\mathcal{M}(S)$, $S = (\mathcal{G}, (\Delta, \mathcal{N}, \mathcal{E}, \mathcal{C}))$ is the first-order formula:*

$$\mathcal{M}(S) = \exists\overline{X}(\bigwedge_{g\in\mathcal{G}} g \wedge \bigwedge_{\delta\in\Delta} \delta \wedge \bigwedge_{\forall\overline{Y_\Gamma}\leftarrow\Gamma\in\mathcal{N}} (\forall\overline{Y_\Gamma} \leftarrow \Gamma.) \wedge \bigwedge_{e\in\mathcal{E}} e \wedge \bigwedge_{c\in\mathcal{C}} c)$$

$\overline{Y_\Gamma}$ is the set of the universally quantified variables in the denial body Γ, and $\overline{X}$ is the set of all the other variables in $\mathcal{M}(S)$.

Definition 3 (ASystem derivation tree). *Given an abductive framework $\langle P$, AB, $IC\rangle$, a query Q and a selection strategy Ξ, an* ASystem derivation tree *is a tree such that:*

- *every node of the tree is an ASystem state.*

- *children nodes are generated by selecting a goal (and if the goal is a denial, further selecting a literal in the denial) according to Ξ, and then applying the inference rules on the selected goal.*
- *the initial state is $S_0 = \langle Q \cup IC, \mathcal{ST}_0 \rangle$, and $\mathcal{ST}_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$.*
- *a success state is one in which $\mathcal{G} = \emptyset$, and $\mathcal{ST}$ is consistent. If $\mathcal{ST}$ is inconsistent or the derivation flounders, then that state is a failure state. A state is a leaf of the tree iff it is either a success or failure state.*

Details of the inference rules and soundness and completeness results are described in [16].

3.2 A Richer Set of Interpretations

According to the definition of our probabilistic semantics (cf. Section 2), every interpretation of abducibles $I \subseteq AB$ has a probability value. This implies that the minimality of abductive solutions, as defined in [13], is no longer a desired property. Since the ASystem incorporates minimality through its rules and interpretations of abducibles, it needs to be modified in order to lift this restriction. To achieve this goal, we will propose an open world interpretation of abducibles via *consistent extended interpretations (CEIs)*. Additionally, since probabilistic inference is currently performed using ground predicates, the ASystem must also be modified such that its success states contain only abducibles, since these are groundable.

The latter modification will be realized using a new safe selection strategy. In order to discuss it, we need to introduce the concept of ASystem types.

Definition 4 (ASystem types). *We distinguish the following* types of atoms *in the abductive context of the ASystem: (i) abducibles (ii) defined predicates and (iii) constraints. The constraint predicates are of the form $X = Y$ and $X \neq Y$ for in/equality constraints, and $X = Y$, $X > Y$, $X < Y$, $\ldots$ for real constraints.*

Given a denial $\forall \overline{Y} \leftarrow \Gamma$, the set Γ of body literals is split into three disjoint sets $\Gamma = ABL \cup NGL \cup OL$. The set ABL contains abducible literals. NGL contains negative non-ground defined predicates and non-ground constraint literals. OL consists of the remaining literals: positive defined literals, negative ground defined literals, and ground constraint literals. Let $\overline{Y}_{NGL}$ denote the set of variables appearing in the elements of NGL and $\overline{Y}_{ABL}$ the set of variables in the elements of ABL.

Definition 5 (Unfolding Safe Selection Strategy). *An* unfolding safe selection strategy *ξ is a safe selection strategy that given the current goal $G = G^- \cup \{\forall \overline{Y} \leftarrow \Gamma\}$ and the selected denial $\forall \overline{Y} \leftarrow \Gamma$, safely selects a literal from Γ in the following manner:*

- *if $OL \neq \emptyset$, select an element from it.*
- *else, if $\overline{Y}_{NGL} \cap \overline{Y}_{ABL} \neq \emptyset$,* (i) *ground all the abducibles containing at least a variable from $\overline{Y}_{NGL} \cap \overline{Y}_{ABL}$;* (ii) *set the new goal to be $G^+ =$*

$G^- \cup \{ground(\forall \overline{Y} \leftarrow \Gamma)\}$, where $ground(\forall \overline{Y} \leftarrow \Gamma)$ is the grounding of the selected denial goal with respect to $\overline{Y}_{NGL} \cap \overline{Y}_{ABL}$ and the non-ground negative abducible literals in NGL, (iii) *apply the unfolding safe selection strategy to the new goal G^+.*

– *else ($OL = \emptyset$ and $\overline{Y}_{NGL} \cap \overline{Y}_{ABL} = \emptyset$), if $ABL \neq \emptyset$, select an element from ABL. If the selected literal is negative and non-ground, we ground it and apply the unfolding safe selection strategy to the newly generated goal.*
– *else fail.*

The new safe selection strategy allows us to prove assumptions on what the denial store $\mathcal{N}$ of any state would contain (see Proposition 1).

Proposition 1. *Given an unfolding safe selection strategy ξ, the denial store in a derivation tree is either empty or its denials have in their body only literals of the following types: abducibles, universally quantified constraints or universally quantified negative literals. Furthermore, for all denials $\forall \overline{Y} \leftarrow \Gamma$ in the denial store it holds that there are no common variables between the abducible literals and the negative non-ground literals and positive non-ground constraint literals ($\overline{Y}_{NGL} \cap \overline{Y}_{ABL} = \emptyset$).*

Example 2. Consider an abductive task with the empty program P, integrity constraints $IC = \{\forall X, Y \leftarrow a(X), not\ p(X), not\ b(Y).\}$ where p is a defined predicate, a is an abducible with domain $\{1, 2\}$ and $b/1$ an abducible with domain $\{3, 4\}$. Applying our system with an empty goal yields a success state in which the denial is moved to the denial store, and nothing is abduced. Given our abducible types and the unfolding selection strategy, our approach first grounds the shared variable X, generating the new goal:

$$\{\forall Y \leftarrow a(1), not\ p(1), not\ b(Y)., \forall Y \leftarrow a(2), not\ p(2), not\ b(Y).\}$$

Let us assume that the first denial is selected as current denial goal[1]. The literal $not\ p(1)$ is selected, and since the predicate p has not definition $not\ p(1)$ succeeds, reducing the goal to $\forall Y \leftarrow a(1), not\ b(Y)$. In this new goal, we can either select $a(1)$, completing the proof with the denial store $\{\forall Y \leftarrow a(1), not\ b(Y).\}$, or we can ground $not\ b(Y)$ to generate the goal $\{\forall Y \leftarrow a(1), not\ b(3)., \forall Y \leftarrow a(1), not\ b(4).\}$.

The unfolding safe selection strategy imposes an important restriction on the denials: the variables that appear both in abducible and non-abducible atoms have a finite domain, according to the domains of the abducible atoms. If one were to lift this restriction, then the denial $\leftarrow a(X), not\ p(X)$. would always fail, assuming that $a(X)$ is an abducible and $p(X)$ is a defined predicate.

Our goal is to have only states whose meanings are *(groundable) formulae containing only abducible predicates*, since the success states and the probability of the abducibles will be used for inference. The unfolding safe selection strategy allows us to remove from the denial store any non-abducible, i.e. according to

[1] The second one is handled similarly.

Proposition 1, universally quantified constraints or negative literals. This is possible due to the second property stated by the proposition claiming that there are no common variables between abducibles and non-abducibles. For example, in a denial such as $\forall Y \leftarrow a, \mathit{not}\ p(Y).$, where a is an abducible and $p(Y)$ is a defined predicate, $p(Y)$ cannot be failed for all Y, so $\mathit{not}\ p(Y)$ is true, and the denial is equivalent to: $\forall Y \leftarrow a$. The same holds for universally quantified constraints in denials. From now on, we assume the denial stores of all states contain only abducibles.

The ASystem interpretations of a success state $S = (\emptyset, (\Delta, \mathcal{N}, \mathcal{E}, \mathcal{C}))$ is simply: $\mathit{ground}(\Delta)$, i.e. all the abducibles in $\mathit{ground}(\Delta)$ are true, and the rest are false. We propose a different understanding of an abductive solution corresponding to a success state, while assuming the use of an unfolding safe selection strategy. The reason we introduce this concept is that the definition of an open world interpretation of abducibles is necessary for correct inference in our probabilistic semantics.

Definition 6 (Consistent Extended Interpretations (CEIs)). *Let S be a success state in the proof of a query Q using an unfolding safe selection strategy and $M(S)$ the meaning of that success state, a ground formula containing only abducibles. The* consistent extended interpretations (CEIs) *of S, denoted by $\mathcal{I}_S$ is the set of models of $M(S)$. Since Δ is part of the conjunction in $M(S)$, all CEIs make the abducibles in Δ true. However, there may be other abducibles which are true in a CEI, hence the title* extended. *These extensions are not arbitrary, instead they must not violate the integrity constraints, encoded in the denial store $\mathcal{N}$, which is part of $M(S)$, hence the title* consistent.

For a query Q, the CEIs $\mathcal{I}_Q$ are simply the union of the all success states, or equivalently, the models of $\bigvee_{i=1}^{n} M(S_i)$, assuming S_i, $i = 1, \ldots, n$ are all the success states for Q.

Changing the perspective on how interpretations of abductive solutions are constructed requires a theoretical justification. Theorem 1 shows that a CEI corresponds to an ASystem interpretation of a success state for an extended query. The extended query is the original query plus the extended part of the CEI, i.e. the abducibles that are true, but not in the abducible store.

Theorem 1. *Consider an abductive framework $\langle P, AB, IC\rangle$ with query Q. Let Δ_i, $i = 1, \ldots, n$ be the abductive solutions. Let $\mathcal{I}_Q$ be the set of consistent extended interpretations of Q. For every $\Delta_i^* \subseteq AB \setminus \mathit{ground}(\Delta_i)$ let $I = \mathit{ground}(\Delta_i) \cup \Delta_i^*$ be a interpretation for the abductive solution to query $Q' = Q \cup \Delta_i^*$ and let $\mathcal{I}_{\Delta^*}$ be the set of all of all such interpretations.*

Then $\mathcal{I}_Q = \mathcal{I}_{\Delta^*}$.

Due to Theorem 1, it is not difficult to extend and prove the notions of soundness and completeness to CEIs.

Theorem 2 (Soundness for CEIs). *Given an abductive framework $\langle P, AB, IC\rangle$ with query Q, and the set $\mathcal{I}_Q$ of consistent extended interpretations, then*

$\forall I \in \mathcal{I}_Q$, $comp_3(P \cup I) \models Q$, and $comp_3(P \cup I)$ is consistent.

Theorem 3 (Completeness for CEIs). *Given an abductive framework $\langle P, AB, IC \rangle$ with query Q, and the set $\mathcal{I}_Q$ of consistent extended interpretations.*
(1) *If $\mathcal{I}_Q = \emptyset$, then $comp_3(P) \models \forall\overline{X}(\neg Q)$; and*
(2) *If $comp_3(P \cup \exists\overline{X}(Q))$ is satisfiable, then $\mathcal{I}_Q \neq \emptyset$.*

Example 3. We illustrate the concept of CEIs through the example of "Friends and Smokers" social network analysis, in the variant presented in the ProbLog 2 tutorial[2], using the standard Prolog syntax. Suppose there are 4 people: person(i), $\forall i = 1, \ldots, 4$ in a social network:

$$\{\text{friend(i,j)} | (i, j) \in \{(1, 2), (2, 1), (2, 4), (3, 2), (4, 2)\}$$

Furthermore, people smoke either because they are stressed, or they are influenced by a friend who smokes, and smoking may cause asthma. We encode this in Prolog as:

```
smokes(X) :- smokes(X, [X]).
smokes(X, _L) :- stress(X).
smokes(X, L) :-
  friend(X,Y),
  \+ member(Y,L),
  influences(Y,X),
  smokes(Y, [Y|L]).
asthma(X) :- smokes(X), smoke_asthma(X).
```

The abducibles in this problem are: stress/1, influences/2, and smoke_asthma/1, where the arguments are of type person.

Assume the query is asthma(1), and the evidence is:
$\{\leftarrow \textit{not smokes}(2).,\leftarrow \textit{influences}(4, 2).\}$. This means we are interested if person 1 has asthma, having observed that person 2 smokes, and person 4 has no influence on person 2. The proof procedure returns four success states, as possible explanations for the query, with the following abducible and denial stores:
$\Delta_1 = \{\text{stress(1), influences(1,2), smoke\_asthma(1), stress(2), influences(2,1)}\}$
$\mathcal{N}_1 = \{\leftarrow \text{influences(4,2)}\}$
$\Delta_2 = \{\text{smoke\_asthma(1), stress(2), influences(2,1)}\}$
$\mathcal{N}_2 = \{\leftarrow \text{influences(4,2)}\}$
$\Delta_3 = \{\text{smoke\_asthma(1), stress(1), influences(1,2)}\}$
$\mathcal{N}_3 = \{\leftarrow \text{influences(4,2)}\}$
$\Delta_4 = \{\text{smoke\_asthma(1), stress(1), stress(2)}\}$
$\mathcal{N}_4 = \{\leftarrow \text{influences(4,2)}\}$

The CEIs for the second success state are the models of the meaning of the state:

$$\text{smoke\_asthma(1)} \land \text{stress(2)} \land \text{influences(2,1)} \land \neg\text{influences(4,2)}$$

[2] `http://dtai.cs.kuleuven.be/problog/v2/tutorial.html#tut_part1_smokers`

instead of the single interpretation $\varDelta_2$. The same holds for the other success states, and the CEIs for the query are the models of the disjunction of all the formulae. This allows the correct inference of the probability of the query, as shown in the next section.

3.3 Probabilistic Inference

The previous subsection has presented the appropriate modification to the ASystem to enable the computation of consistent extended interpretations (CEIs) for a particular success state as the models of the meaning of that success state, and, consequently, for an arbitrary query. The CEIs will be used to compute the probability of a query, given the evidence as integrity constraints. The definition of this quantity is given in Equation 2, Section2. At a first glance, it seems two proofs are necessary, one in order to compute the numerator, using as initial goal the query and the integrity constraints ($Q \cup IC$) and one for the denominator, using as initial goal just the integrity constraints. However, in this manner we prove the integrity constraints twice. To avoid this redundancy, we refine the unfolding safe selection strategy, such that the initial goals and the following subgoals generated by the integrity constraints are solved before the goals and subgoals obtained processing the query[3].

The inference is divided into two parts. The initial goal is $Q \cup IC$ and we stop expanding the proof tree once the integrity constraints are solved. This process ends in *pseudo-success states* of the form: $(Q, (\varDelta, \mathcal{N}, \mathcal{E}, \mathcal{C}))$. To compute the CEIs needed for the denominator in Equation 2, we use the models of $\bigvee_j M(S'_j)$, where $S'_j = (\emptyset, (\varDelta_j, \mathcal{N}_j, \mathcal{E}_j, \mathcal{C}_j))$ is constructed from the pseudo-success state indexed j by eliminating the query Q from the goal.

The second part of the proof, which is needed to compute Equation 3, resumes the application of the inference rules on the partially developed tree from the pseudo-success states (the other leaves are failure states). Finally, we obtain the needed CEIs from the (true) success states.

We discuss the exact probability computation from the meaning of the success states. In our current implementation, we use the idea of ProbLog I [15,4]: we compile $\bigvee_{i=1}^{n} M(S_i)$ (and similarly for $M(S'_j)$ in the case of pseudo-success states) to a BDD, and compute the probability of the BDD.

Example 4. Extending Example 3, suppose that there is 0.3 probability that a person is stressed, 0.4 probability that smoke causes asthma, and 0.2 probability that one friend influences another. Compiling the disjunction of the meaning of the states in a BDD and computing its probability yields the value: 0.2035 as the probability that person 1 has asthma under any explanation.

[3] Note that this refinement concerns the goal selection rather than selecting a literal from a denial, the main feature of an unfolding safe selection strategy.

Furthermore, we can extend the example with more complex forms of evidence. If one observes that in the studied social network, people with asthma don't influence other people to smoke, we can encode this as:
$\leftarrow asthma(X), influences(X, Y)$.

Running the same query after adding this integrity constraint to the program yields a probability of 0.0677, which, as expected, is lower than the probability of person 1 having asthma without this observation. This can be explained also by examining the two success states, with the abducible stores:
$\Delta_1 = \{\text{smoke\_asthma(1)}, \text{stress(2)}, \text{influences(2,1)}\}$
$\Delta_2 = \{\text{stress(1)}, \text{smoke\_asthma(1)}, \text{stress(2)}\}$

These solutions correspond to the second and fourth success states in Example 3. The other two are no longer inferred since in both solutions person 1 influences person 2, and person 1 has asthma, thus violating the newly introduced integrity constraint.

In principle, we could use different approaches to compile and evaluate the ground formulae, such as weighted model counting on DNNFs used in ProbLog 2 [8], or, for approximate inference, the MaxWalkSAT procedure used in Markov logic networks [7].

Probabilistic inference assumes that the probabilities of the abducibles are known. Nevertheless, in many situations, these are not known. Instead, queries or explanations are observed, and the probabilities of the abducibles are learned to maximize the likelihood of the observed data. Based on the encoding of the ground formulae, we can use existing algorithms for parameter learning, e.g. in Section 4 we use the EM algorithm for BDDs proposed in [11] to rank abductive solutions.

4 Evaluation

4.1 Friends and Smokers

In order to scale Examples 3 and 4, we simulate synthetic social networks by generating power law random graphs using Python Web Graph Generator[4]. We vary the maximum nodes from 5 to 200 with a step of 5, and the maximum edges are double the maximum nodes. The obtained graphs are then parsed into appropriate input files for our system, and for ProbLog 2. The initial files contain only one random query atom with predicate asthma, which we enrich with 10 random evidence literals, 5 with the smokers predicate name, and 5 with smokers. We then run the abduction (without BDD compilation and evaluation) and compare our performance with the ProbLog 2 counterpart, the grounding step[5].

[4] `http://pywebgraph.sourceforge.net/`

[5] ProbLog 2 has four steps: grounding, CNF conversion, compilation and evaluation, and our modified ASystem can be used as an alternative to the first step. We run ProbLog 2 with default parameters.

Without evidence, our probabilistic abductive system slightly outperforms grounding on large graphs. This result is expected since our top-down proof grounds only what is needed in the proof of the query, rather than the whole program. In the presence of evidence, however, our current prototype implementation suffers from the lack of tabling, and the time for the proof of the denials increases exponentially in the number of denials. The grounding step of ProbLog 2 has the same complexity when incorporating evidence as in the previous case, since the evidence is treated in a different way[6].

In future work, we plan to improve the runtime of handling integrity constraints by either developing a tabling mechanism for abduction, or solving each integrity constraint separately and assembling the final ground formula as a conjunction of the formulae of the query, resp. of each integrity constraint.

4.2 Gene Interaction

We further evaluate our probabilistic abductive system on the problem of finding network structures in the context of gene interaction networks based on observed data and constraints determined by biological expertise. Our application is motivated by the availability of high-throughput data. The task of analysing such complex data requires computational tools to automatically infer networks from data. Key challenges in network inference include incomplete and noisy input, detection of complex network structures that capture fundamental properties (e.g., robustness oscillations, bistability) of biological systems and computational complexity. An abductive framework caters for constraint checks and prior knowledge incorporation, thus partially dealing with the problems [18].

Our probabilistic abductive system has been used to generate a network of 11 genes, shaped by the nature of the interactions between genes. The different types of interactions between any pair of genes represent our abducibles: $compatible\_regulator(G1, G2, E)$ and $overpowered\_regulator(G2, G2, E2)$ (abbreviated to $r(G1, G2, E)$ and $or(G1, G2, E)$). The first two arguments of these abducibles are genes, whereas the third argument E is a binary variable over the set $\{1, -1\}$ denoting the causal effect of the interaction between two genes. For example, $r(g1, g2, 1)$ (resp. $r(g1, g2, -1)$) means that gene $g1$ activates (resp. inhibits) gene $g2$. Compatible regulators represent regulators that satisfy the sign consistency principle which postulates that the state of a target gene $G2$ is directly related to the state of an activator $G1$ and inversely related to the state of an inhibitor. Overpowered regulators are regulators that are overpowered by a compatible regulator acting on the same target and thus are inconsistent with the sign consistency principle. The probability of the abducibles can be interpreted as the the strength of the knowledge that led to this link being present. The higher the probability the higher the chance that the link is true.

Our perspective on probabilistic abduction as requiring non-minimal solutions is reflected in this experiment as biologists are interested in maximal networks

[6] If our understanding is correct, the truth values of the atoms are propagated in the ground program.

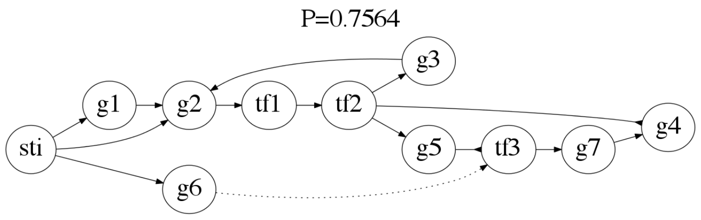

Fig. 1. Normal (resp. dashed) edges are r (resp. *or*) abducibles. Normal (resp. inverted) arrow heads are activation (resp. inhibition).

to distinguish between interactions that are allowed and interactions which are not biologically justified. During the inference process many different instances of the abducibles can be generated and constraints expressing expert knowledge are required to restrict the computation to possible biologically plausible networks. For instance, abducibles have to satisfy existing knowledge of sets of potential gene interactions:

$\leftarrow r(X, Y, E), not\ interactive\_potential(Y, X).$

$\leftarrow or(X, Y, E), not\ interactive\_potential(Y, X).$

It is also important to guarantee that a gene is not assumed to be at the same time a compatible and an overpowered regulator of another gene, and that for each overpowered regulator, there is at least one compatible regulator that can overpower it. These are captured in our model by the constraints: $\leftarrow r(X, Y, E), or(X, Y, E).$

$\leftarrow or(X, Y, E), not\ exists\_overpowered(X, Y).$

$\leftarrow or(X, Y, E), overpowered(Z, X, Y), not\ r(Z, Y, W).$

The biological problem in hand has also insufficient known biological data to provide reliable probabilities on the gene interactions. So instead of applying direct inference, we have used the BDD-based expectation maximization (EM) learning algorithm [11]. Using our probabilistic abductive system we obtain 36 plausible networks. Learning the probabilities of the interaction is done in order to maximize the probability of each network (i.e. the success probability), and this allows the ranking of the networks in terms of their likelihood. We initialize the probabilities to 0.5 and the learning algorithm takes 135 iterations to converge. In the abduced networks the *compatible_regulator* links appear more frequently than *overpowered_regulator*, which is reflected in the learned parameters and, consequently, the ranking of the networks. For example, the top ranked network contains only *compatible_regulator* links. Figure 1 shows a network validated by biological experiments.

A probabilistic abductive framework such as the one proposed in this paper extends the benefits of abductive inference to capturing noise in the input data and dealing with the problem of model selection. Given the number of variables involved, there are a vast number of possible network topologies, and the problem of model selection is combinatorial. Validating each of them would far exceed practical resources. Our probabilistic approach provides preference measures over links within a network and over networks, thus helping in the design of follow up experiments to discriminate between models.

5 Related Work

Logical-probabilistic reasoning has been defined in the context of deduction, induction and abduction. In this section, we compare our approach to existing work that focuses on probabilistic abduction or uses distribution semantics.

Probabilistic Horn abduction, later developed into the independent choice logic (ICL) [19], is one of the first probabilistic abduction frameworks. ICL accepts as input normal logic programs, but does not support integrity constraints. PRiSM [21] is a system created by the authors of distribution semantics that allows negation as failure, yet probabilistic abduction in PRiSM does not handle integrity constraints, and explanations are required to be mutually exclusive. ProbLog [5,15] is defined in a deductive setting, so it does not feature integrity constraints, and furthermore, the use of negation is limited to probabilistic facts or predicates which are not defined based on probabilistic facts. The latter issue has been addressed in ProbLog 2 [8], a system which, similarly to answer set programming, relies on grounding. Our system postpones grounding as much as possible, and the defined predicates in the program are not required to be groundable. Probabilistic abduction has also been defined in the context of constraint-handling rules (CHR) systems [1] where clauses are definite, and the integrity constraints can contain only abducibles. A probabilistic abduction method for classical negation which allows the encoding of integrity constraints is introduced in [11], but does not propose a probabilistic semantics in an abductive setting.

Markov logic networks (MLNs) [7] provide a different framework for combining probabilistic and logical reasoning. Markov networks are used as a probabilistic model and first-order theories encode the knowledge. Our approach is significantly different in the sense that it is based on abductive logic programming and the distribution semantics. In a MLN all formulas are treated as soft constraints, while in our approach we clearly distinguish between the rules in the logic program and the integrity constraints expressed as denials. Furthermore, we treat the integrity constraints as hard constraints: the consistent extended interpretations never violate the constraints. The possibility of viewing denials as soft constraints is a direction we wish to pursue in future work.

6 Conclusions

In this paper, we have proposed a method for applying distribution semantics in an abductive framework and provided an implementation based on the abductive procedure ASystem, showing how it can be adapted for probabilistic inference by removing the requirement that an abductive solution should be minimal. We have formally shown that our framework is correct with respect to distribution semantics. Advantages of our approach include the ability to handle negation as failure and integrity constraints as denials, as well as numerical constraints and term (in-)equality. We have evaluated our implementation by applying it to the task of finding biologically plausible networks describing gene interactions, which requires discovering non-minimal solutions.

Future work includes making probabilistic inference in our system feasible for larger problems through improved efficiency by tabling mechanisms. Due to the similarity between abduction and induction, the former can be used as an inference mechanism in inductive logic programming (ILP). We plan to integrate our probabilistic abduction framework in a probabilistic ILP context, building on results in [3] where the ASystem is used to explore the hypothesis space in order to find solutions to an ILP task.

References

1. Christiansen, H.: Implementing Probabilistic Abductive Logic Programming with Constraint Handling Rules. In: Schrijvers, T., Frühwirth, T. (eds.) Constraint Handling Rules. LNCS (LNAI), vol. 5388, pp. 85–118. Springer, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-92243-8_5
2. Clark, K.L.: Negation as failure. In: Logic and Data Bases, pp. 293–322 (1977)
3. Corapi, D., Russo, A., Lupu, E.: Inductive logic programming as abductive search. In: ICLP (Technical Communications), pp. 54–63 (2010)
4. De Raedt, L., Kimmig, A., Gutmann, B., Kersting, K., Santos Costa, V., Toivonen, H.: Probabilistic inductive querying using ProbLog. Springer (2010), https://lirias.kuleuven.be/handle/123456789/284080
5. De Raedt, L., Kimmig, A., Toivonen, H.: ProbLog: A probabilistic Prolog and its application in link discovery. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007, pp. 2462–2467 (2007)
6. Denecker, M.: Knowledge representation and reasoning in incomplete logic programming. Ph.D. thesis, Department of Computer Science, K.U.Leuven, Leuven, Belgium, de Schreye, Danny and Denef, Jan (supervisors) (September 1993), https://lirias.kuleuven.be/handle/123456789/131431
7. Domingos, P., Kok, S., Poon, H., Richardson, M., Singla, P.: Unifying logical and statistical ai. In: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI 2006, vol. 1, pp. 2–7. AAAI Press (2006), http://dl.acm.org/citation.cfm?id=1597538.1597540
8. Fierens, D., den Broeck, G.V., Thon, I., Gutmann, B., Raedt, L.D.: Inference in probabilistic logic programs using weighted cnf's. In: Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI 2011), pp. 211–220. AUAI Press, Corvallis (2011)
9. Fitting, M.: A kripke-kleene semantics for logic programs. J. Log. Program. 2(4), 295–312 (1985)
10. Fung, T.H., Kowalski, R.: The iff proof procedure for abductive logic programming. The Journal of Logic Programming 33(2), 151–165 (1997), http://www.sciencedirect.com/science/article/pii/S0743106697000265
11. Inoue, K., Sato, T., Ishihata, M., Kameya, Y., Nabeshima, H.: Evaluating abductive hypotheses using an em algorithm on bdds. In: Proceedings of the 21st International Jont Conference on Artifical Intelligence, IJCAI 2009, pp. 810–815. Morgan Kaufmann Publishers Inc., San Francisco (2009), http://dl.acm.org/citation.cfm?id=1661445.1661574

12. Kakas, A., Michael, A., Mourlas, C.: Aclp: Abductive constraint logic programming. The Journal of Logic Programming 44(1-3), 129–177 (2000), `http://www.sciencedirect.com/science/article/pii/S0743106699000758`
13. Kakas, A.C., Kowalski, R.A., Toni, F.: Abductive logic programming. J. Log. Comput. 2(6), 719–770 (1992)
14. Kakas, A.C., Van Nuffelen, B., Denecker, M.: A-system: Problem solving through abduction. In: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, vol. 1, pp. 591–596. IJCAI, inc. and AAAI, Morgan Kaufmann Publishers, Inc. (2001), `http://www.cs.kuleuven.ac.be/cgi-bin-dtai/publ_info.pl?id=34862`
15. Kimmig, A.: A Probabilistic Prolog and its Applications. Ph.D. thesis, K.U. Leuven (2010)
16. Ma, J.: Distributed Abductive Reasoning: Theory, Implementation and Application. Ph.D. thesis, Imperial College London (2011)
17. Ma, J.: Abductive reasoning module for sicstus prolog (2012), `http://www-dse.doc.ic.ac.uk/cgi-bin/moin.cgi/abduction`
18. Maimari, N., Broda, K., Kakas, A., Krams, R., Russo, A.: Arni: Abductive inference of complex regulatory network structures (poster). In: 21st Annual International Conference on Intelligent Systems for Molecular Biology, 12th European Conference on Computational Biology, ISMB/ECCB 2013 (2013)
19. Poole, D.: Abducing through negation as failure: stable models within the independent choice logic. J. Log. Program. 44(1-3), 5–35 (2000)
20. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: Proceedings of the 12th International Conference on Logic Programming, pp. 715–729. MIT Press (1995)
21. Sato, T., Kameya, Y.: Parameter learning of logic programs for symbolic-statistical modeling. J. Artif. Intell. Res. (JAIR) 15, 391–454 (2001)
22. Sato, T., Kameya, Y., Zhou, N.F.: Generative modeling with failure in prism. In: IJCAI, pp. 847–852 (2005)