Integrating Logical Reasoning and Probabilistic Chain Graphs

Arjen Hommersom, Nivea Ferreira, and Peter J.F. Lucas

Institute for Computing and Information Sciences,
Radboud University Nijmegen, Nijmegen, The Netherlands
{arjenh,nivea,peterl}@cs.ru.nl

Abstract. Probabilistic logics have attracted a great deal of attention during the past few years. While logical languages have taken a central position in research on knowledge representation and automated reasoning, probabilistic graphical models with their probabilistic basis have taken up a similar position when it comes to reasoning with uncertainty. The formalism of chain graphs is increasingly seen as a natural probabilistic graphical formalism as it generalises both Bayesian networks and Markov networks, and has a semantics which allows any Bayesian network to have a unique graphical representation. At the same time, chain graphs do not support modelling and learning of relational aspects of a domain. In this paper, a new probabilistic logic, chain logic, is developed along the lines of probabilistic Horn logic. The chain logic leads to relational models of domains in which associational and causal knowledge are relevant and where probabilistic parameters can be learned from data.

1 Introduction

There has been a considerable amount of work in the field of artificial intelligence during the past two decades on integrating logic and probability theory. This research was motivated by perceived limitations of both formalisms. Logic has for long acted as the common ground for almost all research on knowledge representation, reasoning and learning in artificial intelligence; yet, uncertainty cannot be handled easily in logic. Probabilistic graphical models take probability theory as their foundation; they have been proposed as formalisms for statistical learning and for reasoning with uncertainty. Although their associated graphical representation allows specifying relationship among objects in the domain of discourse such that it is possible to reason about their statistical dependences and independences, probabilistic graphical models are essentially propositional in nature, and they lack the representational richness of logics.

Several researchers have proposed probabilistic logics that merge those two types of languages in an attempt to redress their individual shortcomings. A variety of such languages is now available, each of them adopting a different view on the integration. Unfortunately, it appears that all of the available frameworks are still restricted in one way or the other. In particular, the available languages either support representing Bayesian-network-like independence information or Markov-network-like independence information. In this paper, we describe a probabilistic first-order language that is more expressive than similar languages developed earlier, in the sense that the probabilistic models that can be specified and reasoned about have Bayesian and Markov networks as special cases. This new probabilistic logic is called *chain logic*. This paper addresses the representation and reasoning aspects of chain logic as well as parameter learning of chain logic theories.

The organisation of this paper is as follows. In Section 2 we provide an overview of the basic notions of Horn clauses and chain graphs. Section 3 contains an introduction to the chain logic language, with details on its syntax and semantics. In Section 4, we focus on learning the parameters of chain logic theories. In Section 5 the most important related work is introduced and a detailed comparison to this is provided. Finally, Section 6 presents our conclusions.

2 Preliminaries

The work discussed in this paper builds upon two separate branches of research: (i) probabilistic graphical models, and (ii) abductive logic. We start by summarising the basic facts about probabilistic graphical models, in particular chain graph models. This is followed by a review of central notions from abductive logic. Both frameworks act as the foundation for chain logic as developed in the remainder of the paper.

2.1 Chain Graphs

A chain graph (CG) is a probabilistic graphical model that consists of labelled vertices, that stand for random variables, connected by directed and undirected edges. This representation allows chain graphs to be considered as a framework that generalises both directed acyclic graph probabilistic models, i.e., Bayesian networks, and undirected graph probabilistic models, i.e., Markov networks [4]. The definitions with respect to chain graphs given in this paper are in accordance with [5].

Let $G = (V, E)$ be a *hybrid graph*, where V denotes the set of *vertices* and E the set of *edges*, where an edge is either an *arc* (directed edge), or a *line* (undirected edge). Let indexed lower case letters, e.g., v_1 and v_2, indicate vertices of a chain graph. We denote an arc connecting two vertices by '$\rightarrow$' and a line by '$-$'. Consider two vertices v_1 and v_2. If $v_1 \rightarrow v_2$ then v_1 is a *parent* of v_2. If $v_1 - v_2$ then v_1 (v_2) is a *neighbour* of v_2 (v_1). The set of parents and neighbours of a vertex v are denoted by $\mathrm{pa}(v)$ and $\mathrm{ne}(v)$, respectively.

A *path* of length n in a hybrid graph $G = (V, E)$ is a sequence of distinct vertices $v_1, \ldots, v_{n+1}$, such that either $v_i - v_{i+1} \in E$, $v_i \rightarrow v_{i+1} \in E$, or $v_i \leftarrow v_{i+1} \in E$. A *directed path* is a path which includes at least one arc, and where all arcs have the same direction. A *cycle* is a path where the first and last vertex are the same. A *chain graph* is a hybrid graph with the restriction that no directed cycles exist.

If there is a line between every pair of vertices in a set of vertices, then this set is named *complete*. A *clique* is a maximally complete subset. Now, consider the graph obtained from a chain graph by removing all its arcs. What are left are vertices connected by lines, called *chain components*; the set of all chain components is denoted here by $\mathcal{C}$.

Associated to a chain graph $G = (V, E)$ is a joint probability distribution $P(X_V)$ that is faithful to the chain graph G, i.e., it includes all the independence information represented in the graph. This is formally expressed by the following *chain graph Markov property*:

$$P(X_V) = \prod_{C \in \mathcal{C}} P(X_C \mid X_{\mathrm{pa}(C)}) \tag{1}$$

with $V = \bigcup_{C \in \mathcal{C}} C$, and where each $P(X_C \mid X_{\mathrm{pa}(C)})$ factorises according to

$$P(X_C \mid X_{\mathrm{pa}(C)}) = Z^{-1}(X_{\mathrm{pa}(C)}) \prod_{M \in M(C)} \varphi_M(X_M) \tag{2}$$

given that $M(C)$ is the complete set in the moral graph[1] obtained from the subgraph $G_{C \cup \mathrm{pa}(C)}$ of G. The functions φ are non-negative real functions, called *potentials*; they generalise joint probability distributions in the sense that they do not need to be normalised. Finally, the normalising factor Z is defined as

$$Z(X_{\mathrm{pa}(C)}) = \sum_{X_C} \prod_{M \in M(C)} \varphi_M(X_M) \tag{3}$$

As a Bayesian network is a special case of a chain graph model, Equation (1) simplifies in that case to:

$$P(X_V) = \prod_{v \in V} P(X_v \mid X_{\mathrm{pa}(v)}) \tag{4}$$

which is the well-known factorisation theorem of Bayesian networks [5]. In this case, the chain components are formed by a family of random variables. Therefore, for each of those random variables the distribution is defined as the conditional probability function of this variable, given the value of its parents. Note that according to Equation (1), chain graphs can also be interpreted as a directed acyclic graph of chain components.

2.2 Abduction Logic

Abduction logic is defined as a special variant of function-free Horn logic, where the syntax of Horn clauses is slightly modified, and logical implication, '←', is given a causal interpretation. *Abduction clauses* have the following form:

$$D \leftarrow B_1, \ldots, B_n : R_1, \ldots, R_m$$

[1] Moralisation encompasses: (1) adding lines between unconnected parents of a chain component, and (2) conversion of arcs into lines by ignoring their directions.

where the predicates of the atoms D and B_i are at least unary and the atoms R_j, called *templates*, express relationships among variables, where at least one variable appearing in the atoms D and B_i occurs in at least one template R_j. An example illustrating this representation is shown below (Example 1). Atoms that do not occur as head of a clause are called *assumables*. From a logical point of view, the ':' operator has the meaning of a conjunction; it is only included in the syntax to allow separating atoms that are templates from non-template atoms. The basic idea is to use atoms D and B_i to introduce specific variables, later interpreted as *random* variables, and the templates R_j to represent relations among those variables. Other variables can be introduced to define additional, logical relationships among objects, or to define generic properties.

Let T be a set of abduction clauses, called an *abductive theory* in this paper. Then, concluding a formula ψ from the theory is denoted by $T \vDash \psi$ (when using model theory) and $T \vdash \psi$ (when using deduction or proof theory).

Throughout this paper, we will write Ψ' as the set of ground instances of Ψ, where Ψ is a set of formulae. For example, $\mathcal{A}$ is the set of all assumables and we use $\mathcal{A}'$ to denote the set of ground instances of $\mathcal{A}$.

For abduction logic a special type of logical reasoning has been proposed, called *abduction*, which is defined in terms of model theory or deduction using so-called *explanations*: "a entails b" allows inferring a as an explanation of b. Given a set of atoms O, interpreted as *observations*, then these observations are explained in terms of the abductive theory and a set of assumables.

Definition 1. *An* explanation *of a set of atoms O based on the pair $\langle T, \mathcal{A}\rangle$ is defined as a set of ground assumables $E \subseteq \mathcal{A}'$ satisfying the following conditions:*

- *$T \cup E \vDash O$, and*
- *$T \cup E$ is consistent, i.e., $T \cup E \nvDash \bot$.*

A minimal explanation *E of O is an explanation whose proper subsets are not explanations of O. The set of all minimal explanations is denoted by $\mathcal{E}_T(O)$.*

Example 1. Suppose that we have the following piece of medical knowledge. Influenza (I) causes coughing (C), where coughing is known as a possible cause for hoarseness (H). In addition, coughing is known to be associated with dyspnoea (shortness of breath) (D), although a clear cause-effect relationship is missing. Dyspnoea restricts the oxygen supply to the blood circulation; the resulting low oxygen saturation of the blood will turn the skin to colour blue (B), which is a condition called cyanosis. This qualitative knowledge is represented by the causal network shown in Fig. 1. The associated abductive theory T is the following:

$$
\begin{aligned}
&I(x) \leftarrow: r_I(x)\\
&C(x) \leftarrow I(y) : r_{C,I}(x,y), r_{C,D}(x,z)\\
&D(x) \leftarrow I(y) : r_{C,I}(z,y), r_{C,D}(z,x)\\
&H(x) \leftarrow C(y) : r_{H,C}(x,y)\\
&B(x) \leftarrow D(y) : r_{B,D}(x,y)
\end{aligned}
$$

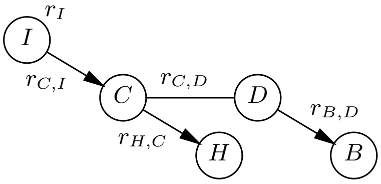


Fig. 1. Causal network model of causal and associational knowledge about influenza.

where each of the variables has $\{f, t\}$ as domain. It now holds that:

$$T \cup \{r_I(t), r_{H,C}(t,t), r_{C,I}(t,t), r_{C,D}(t,t)\} \vDash H(t)$$

and $T \cup \{r_I(t), r_{H,C}(t,t), r_{C,I}(t,t), r_{C,D}(t,t)\} \nvDash \bot$.

The intuition behind the syntax of abduction clauses, such as $C(x) \leftarrow I(y) : r_{C,I}(x,y), r_{C,D}(x,z)$, is that $C(x) \leftarrow I(y)$ expresses the *potential* existence of a causal relation between the referred atoms, here $I(y)$ and $C(x)$. Note that $I(y)$ also appear in the clause $D(x) \leftarrow I(y) : r_{C,I}(z,y), r_{C,D}(z,x)$, following the fact that I is the parent of the chain component CD. Templates R_j, e.g. $r_{C,I}(x,y)$, expresses whether the relationship actually does or does not hold. When there are no atoms to the left of the ':' operator, such as in the clause $I(x) \leftarrow: r_I(x)$, the template represents a root node or an association with no parents.

3 Chain Logic

In this section, the chain logic language is formally defined. This paves the way for the next section where we will focus on learning.

3.1 Language Syntax

The formalism presented in this section is inspired by probabilistic Horn logic as introduced by Poole in [1]. For the sake of simplicity, we assume here finite domain specifications (infinite domains are briefly mentioned in Section 6). Furthermore, the unique names assumption holds for the different constants of the domain.

Chain logic (CL) extends abduction logic as described in Section 2.2 by interpreting templates as representing uncertain events. The actual definition of the uncertainty is done by means of a *weight* declaration. This is of the form

$$weight(a_1 : w_1, \ldots, a_n : w_n) \tag{5}$$

where a_i represents an atom and $w_i \in \mathbb{R}_0^+$. The set of atoms appearing in such declarations are the assumables $\mathcal{A}$. Here we require that the atoms in a weight declaration share the same variables. Furthermore, we require that a ground atom a – which is an instance of one of the assumables – does not appear as an instance of another assumable in another weight declaration. The weight declaration defines conjunctions of atoms that are mutually exclusive and exhaustive. Therefore, together with the above elements, a CL specification also includes integrity constraint statements, i.e., clauses of the form

$$\bot \leftarrow a_i, a_j \tag{6}$$

for any pair a_i and a_j appearing in the same weight declaration where $i \neq j$. Such clauses are implicit in all of our given examples. We also allow the addition of another set of constraints referring to a pair of assumables appearing in different weight declarations, as seen in the example below.

Example 2. Consider the description given in Example 1. Uncertainty is defined by replacing the templates by potential functions. For the abductive theory in this example:

φ_{CI}	$\imath$	$\bar{\imath}$
c	8	2
$\bar{c}$	1	10

φ_{CD}	d	$\bar{d}$
c	18	2
$\bar{c}$	5	2

φ_{HC}	c	$\bar{c}$
h	0.6	0.1
$\bar{h}$	0.4	0.9

φ_{BD}	d	$\bar{d}$
b	0.3	0.001
$\bar{b}$	0.7	0.999

φ_{I}	
$\imath$	0.1
$\bar{\imath}$	0.9

This example can be represented in chain logic using the following abduction clauses:

$$\begin{array}{l}
I(x) \leftarrow: \varphi_I(x) \\
C(x) \leftarrow I(y) : \varphi_{CI}(x, y), \varphi_{CD}(x, z) \\
D(x) \leftarrow I(y) : \varphi_{CI}(z, y), \varphi_{CD}(z, x) \\
H(x) \leftarrow C(y) : \varphi_{HC}(x, y) \\
B(x) \leftarrow D(y) : \varphi_{BD}(x, y) \\
\bot \leftarrow \varphi_{CI}(x, y), \varphi_{CD}(\bar{x}, z)
\end{array}$$

Furthermore, we can associate weights to the assumables according to the potential functions. For instance,

$$\mathit{weight}(\varphi_{CD}(t, t) : 18, \varphi_{CD}(t, f) : 2, \varphi_{CD}(f, t) : 5, \varphi_{CD}(f, f) : 2)$$

In order to be able to probabilistically interpret a CL theory T, a number of assumptions are added to those of abduction logic: (i) the theory is acyclic; (ii) the rules for every ground non-assumable represented in T' are covering, i.e., there is always a rule whose assumable holds; (iii) the bodies of the rules in T' for an atom are mutually exclusive; (iv) there is a set of ground assumables, one from every grounded weight declaration, consistent with T. As in Poole's probabilistic Horn logic, these assumptions are not intended to be enforced by the system: it is up to the modeller to comply to these requisites. Under this condition, we can then guarantee the probabilistic properties of the theory.

3.2 Semantics and Reasoning

The interpretation of chain logic theories T is done in terms of possible world semantics for the ground case.

Definition 2. *Let $\mathcal{P}$ be a set of predicates of the language. Then a possible world is a tuple $w = \langle D, \omega, \hat{p} \rangle$ where*

- *D is a set of ground terms of the language*
- *$\omega : \mathcal{A}' \to R_0^+$ is a function which assigns a weight to ground assumables $\mathcal{A}'$*
- *$\hat{p} : D^n \to \{true, false\}$ is a valuation function, for each $p \in \mathcal{P}$.*

Truth of formulae is then inductively defined as usual except that an atom a is false if there are clauses $a \leftarrow b_1$ and $a \leftarrow b_2$ in T and both b_1 and b_2 are true. Furthermore, we have:

$$\begin{array}{rl} w \models & weight(a_1 : w_1, \ldots, a_n : w_n) \\ \textit{iff} & \exists i\, w \models a_i \text{ and } \forall j \neq i\, w \not\models a_j \text{and } \forall i\, \omega(a_i) = w_i \end{array}$$

which expresses that exactly one assumable is true in a weight declaration.

As a convenience, for arbitrary theories, we write $w \models T$, whenever for all groundings of T, denoted by T', we have $w \models T'$. The set of all possible worlds denoted W, for which we define a joint probability distribution.

Definition 3. *Let P_T be a non-negative real function of W that is defined as follows:*

$$P_T(w) = \begin{cases} \frac{1}{Z} \prod_{a \in \mathcal{A}'} \omega(a) & \textit{if } w \models T \\ 0 & \textit{otherwise} \end{cases}$$

where $Z = \sum_{w \in \{w | w \models T\}} \prod_{a \in \mathcal{A}'} \omega(a)$.

Clearly, the function P_T obeys the axioms of probability theory, as each weight is larger than or equal to 0 and, given that there is a set of consistent assumables consistent with T, there is at least one possible world for T, thus, it follows that $\sum_{w \in W} P_T(w) = 1$. Therefore, it is a joint probability distribution; P_T is sometimes abbreviated to P in the following. A probability for a formula conjunction φ can be derived by marginalising out the other atoms, i.e., $P(\varphi) = \sum_{w \models \varphi} P(w)$.

These definitions provide the means to reason logically, at the same time assigning probabilities to conjunctive formulae in the language. An alternative way to look at the reasoning process, however, is in terms of explanations of observations, as defined above, which will be considered next.

We define a *hypothesis* as a conjunction of ground instances, one for each assumable of a grounded weight declaration. The set of all such hypotheses is denoted by $\mathcal{H}$. The set of consistent hypotheses, with respect to T will be denoted by CH, i.e., $\mathrm{CH} = \{H \in \mathcal{H} \mid T \cup H \not\models \perp\}$.

Proposition 1. *The joint probability distribution over the set of hypotheses is as follows:*

$$P_T(H) = \begin{cases} \frac{1}{Z} \prod_{a \in H} \omega(a) & \textit{if } H \in \mathrm{CH} \\ 0 & \textit{otherwise} \end{cases}$$

where $Z = \sum_{H \in \mathrm{CH}} \prod_{a \in H} \omega(a)$.

Given T, a minimal explanation E of some formula ψ is equivalent to a disjunction of hypotheses, i.e., $E \equiv \bigvee_i H_i$ with $H_i \in \mathcal{H}$. As all H_i are mutually exclusive, it follows that:

$$P_T(E) = P_T(\bigvee_i H_i) = \sum_i P_T(H_i)$$

which assigns a probability to minimal explanations. In order to assign a probability to a formula using explanations, we have the following result.

Theorem 1. *Under the assumptions mentioned in Section 3.1, if $\mathcal{E}_T(\psi)$ is the set of minimal explanations of the conjunction of atoms ψ from the chain logic theory T, then:*

$$P_T(\psi) = \sum_{E \in \mathcal{E}_T(\psi)} P(E)$$

Proof. This follows exactly the same line of reasoning of [1, page 53, proof of Theorem A.13]. □

This result shows that P is indeed a probability distribution over conjunctions of formulae if we use the definition of P_T above. Other probabilities can be calculated on the basis of these types of formulae, such as conditional probabilities. Below, we will sometimes refer to the resulting probability distribution by P_T in order to stress that we mean the probability calculated using Definition 3.

Example 3. Reconsider the uncertainty specification concerning influenza as described in Example 2. Consider here that we are interested in calculating the $P(B(t))$ (i.e., the probability of B being true). Recalling the definitions provided in Section 2.2, we obtain the minimal explanations for $B(t)$, i.e., $\mathcal{E}_T(B(t))$ as the set with the (8) members such as:

$$\begin{gathered}\{\varphi_{BD}(t,t), \varphi_{CD}(t,t), \varphi_{CI}(t,t), \varphi_I(t)\} \\ \{\varphi_{BD}(t,t), \varphi_{CD}(t,t), \varphi_{CI}(t,f), \varphi_I(f)\} \\ \vdots\end{gathered}$$

We can then sum over the hypotheses that are consistent with these explanations.

$$\begin{aligned}P(B(t)) &= \textstyle\sum_{e \in \mathcal{E}_T(B(t))} P(e) \\ &= Z^{-1}(0.3 \cdot 18 \cdot 8 \cdot 0.1 + \ldots) = 27.7/Z\end{aligned}$$

Similarly, we can find that $P(B(f)) = 88.0/Z$, so $Z = 115.7$ and thus $P(B(t)) \approx 0.24$.

Abductive reasoning establishes the relevant variables for the computation of a marginal probability, i.e., it selects the portion of the chain graph that is relevant. Consider once again our running example. By asking if B is true, we obtain through the rule $B(x) \leftarrow D(y) : \varphi_{BD}(x,y)$ that D influences B. In terms of the graph, this means that we walk the arc in reverse direction, i.e., from

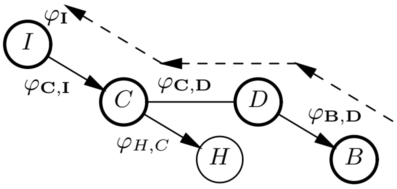


Fig. 2. The direction of reasoning about B is denoted with a dashed line. The nodes that represent variables that are abduced over and the variables that are relevant in the explanation are highlighted. Parts of the graph that are not relevant for the computation of $P(B)$ are not considered.

effect to explaining cause, from B to D. We now look at the rules which have D as head, selecting $D(x) \leftarrow I(y) : \varphi_{CI}(z, y) \wedge \varphi_{CD}(z, x)$. From the potential φ_{CD} in this clause, the existence of the association between C and D is established. From the presence of potential φ_{CI} we – indirectly – recover and include also the influence of I on C. From the presence of predicate $I(y)$ we can proceed to including also the potential φ_I, which (as seen previously) is also important for the correct probabilistic computation. This process is graphically depicted in Fig. 2.

Example 4. Consider that we are interested in the probability of $P(I(t) \mid B(t))$. This probability can be obtained by the having $P(I(t) \wedge B(t))$ divided by $P(B(t))$. The calculation of $P(B(t))$ was shown in Example 3. By calculation the minimal explanations for $I(t) \wedge B(t)$, we obtain that $P(I(t) \wedge B(t)) = 4.5/Z \approx 0.04$, so it follows that $P(I(t) \mid B(t)) \approx \frac{0.04}{0.24} \approx 0.16$. Note that the prior probability for $I(t)$ is 0.1, so the evidence $B(t)$ has increased the probability for influenza.

3.3 Specification of Chain Graphs

In this section, we present the formal relation between chain graphs with discrete random variables and chain logic. For the sake of simplicity, we focus on chain graphs with binary variables, i.e., the set of constants is $\{t, f\}$, although the theory generalises to arbitrary arities. Complementary constants are denoted with a bar, i.e., $\bar{t} = f$ and $\bar{f} = t$.

The translation from a chain graph G to a chain logic theory T is as follows. First, introduce for each potential function φ_M a corresponding predicate φ_M and define a weight declaration containing $\varphi_M(c_0, \ldots, c_n) : w$ if $\varphi_M(X_M = (c_0, \ldots, c_n)) = w$, for all possible instantiations $(c_0, \ldots, c_n)$ of X_M. Second, we model the structure of the chain graph in chain logic. Consider a vertex v in G. For each component $C \in \mathcal{C}$ of G, there is a set of potential functions defined on the moral graph of the sub-graph $G_{C \cup \mathrm{pa}(C)}$ which contains v or one of the parents of C. This set of potential functions is denoted by $\Phi_G(C, v)$. For every vertex v, we have the following formula in T:

$$\begin{aligned} V(x) \leftarrow \bigwedge\{V'(x_{v'}) \mid v' \in \mathrm{pa}(C)\} :& \\ \bigwedge\{\varphi_M(x_1, \ldots, x, \ldots, x_n) \mid \varphi_M \in \Phi_G(C, v)\}& \end{aligned}$$

and we ensure that each of the predicates defined for the same random variable shares that variable in the formula. However, this is not strictly necessary as different values for the same random variable in a component is also disallowed by the integrity constraints.

The integrity constraints are defined as follows. If we have two potential functions, namely an n-ary $\varphi_M(\ldots, v, \ldots)$ and an m-ary $\varphi'_M(\ldots, v, \ldots)$, i.e., which share a variable v *in the same chain component* (i.e., not between chain components), then we add the following formula to T:

$$\bot \leftarrow \varphi_M(x_0, \ldots, x, \ldots, x_n),\ \varphi'_M(x'_0 \ldots, \bar{x}, \ldots, x'_m)$$

for each variable that they share. As mentioned earlier, this ensures we do not generate explanations which have inconsistent assignments to the random variables within the same chain component.

In the following theorem, we establish that probabilities calculated from the chain logic theory correspond to the chain graph semantics.

Theorem 2. *Suppose $v_1, \ldots, v_n$ are vertices in a chain graph, with T as the corresponding chain logic theory by the translation described above, then:*

$$P(X_{v_1} = c_1, \ldots, X_{v_n} = c_n) = P_T(V_1(c_1), \ldots, V_n(c_n))$$

Proof. There is only one minimal explanation of $V_1(c_1) \wedge \cdots \wedge V_n(c_n)$, namely $\varphi_M(c_0^M, \ldots, c_m^M)$ for all potential functions in cliques in the moral graphs of chain components with their parents, such that the constants filled into the potential functions correspond to the values for each of the random variables.

The explanation describes exactly one hypothesis. Denote this as h. As the potential functions are related to exactly one component, we have the following equation:

$$\begin{aligned} & \prod_{a \in h} \omega(a) \\ & = \prod_{C \in \mathcal{C}} \prod_{\varphi_j^C(c_0^j, \ldots, c_n^j) \in h} \varphi_j^C(X_{v_0^j} = c_0^j, \ldots, X_{v_n^j} = c_n^j) \\ & = \prod_{C \in \mathcal{C}} \prod_{M \in M(C)} \varphi_M(X_M) \end{aligned} \tag{7}$$

where φ^C are potential functions defined for component C and $M(C)$ are the complete sets in the moral graph from the sub-graph $G_{C \cup \mathrm{pa}(C)}$.

Let $Z = \sum_{H \in \mathrm{CH}} \prod_{a \in H} \omega(a)$. Since there are no integrity constraints between variables in chain components (i.e., combinations of consistent potential functions which are in different chain components are consistent), we have that:

$$\begin{aligned} Z &= \sum_{H \in \mathrm{CH}} \prod_{a \in H} \omega(a) \\ &= \prod_{C \in \mathcal{C}} \sum_{h \in \mathrm{CH}(C)} \prod_{\varphi_j^C} \varphi_j^C(X_{v_0^j} = c_0^j, \ldots, X_{v_n^j} = c_n^j) \\ &= \prod_{C \in \mathcal{C}} Z(X_{\mathrm{pa}(C)}) \end{aligned} \tag{8}$$

where CH(C) is the set of consistent hypotheses (w.r.t. T) restricted to the potential functions in that chain component. Then, the equivalence follows in the following way:

$$\begin{aligned} & P(X_{v_1} = c_1, \ldots, X_{v_n} = c_n) \\ & =_{\text{(factorisation)}} \textstyle\prod_{C \in \mathcal{C}} P(X_C \mid X_{\mathrm{pa}(C)}) \\ & =_{\text{(factorisation)}} \textstyle\prod_{C \in \mathcal{C}} Z^{-1}(X_{\mathrm{pa}(C)}) \prod_{M \in M(C)} \varphi_M(X_M) \\ & \quad \times \left(\textstyle\prod_{C \in \mathcal{C}} Z^{-1}(X_{\mathrm{pa}(C)})\right) \prod_{C \in \mathcal{C}} \prod_{M \in M(C)} \varphi_M(X_M) \\ & =_{\text{(Eq. 8)}} Z^{-1} \textstyle\prod_{C \in \mathcal{C}} \prod_{M \in M(C)} \varphi_M(X_M)) \\ & =_{\text{(Eq. 7)}} Z^{-1} \textstyle\prod_{a \in w} \omega(a) \\ & =_{\text{(def. } P_T)} P_T(V_1(c_1), \ldots, V_n(c_n)) \end{aligned}$$

□

As we have shown in Section 3.2 that P_T adheres to the axioms of probability theory, chain graphs and the translated chain logic theory agree on all probabilities. This result shows that chain graphs can be translated to chain logic specifications. The converse is also true: all chain logic theories, which adhere to the assumptions of Section 3.1, correspond to a chain graph as a fully connected Markov network models and the associated probability distributions. This is not a minimal independence map of the underlying probability distribution in general, although conditional independence statements can be obtained by comparing explanations between formulae. As we are only able to represent direct causal links and indirect association, we conjecture that we have the same expressiveness in this logic as in chain graphs.

4 Learning Chain Graph Parameters

While observables and assumables make up the core of chain logic, determining the probabilistic parameters of assumables using observations stored in a database D is one of the essential tasks of learning in chain logic. Basically, the goal is to estimate weights of the assumables in a CL theory related to CGs. In general, it is not possible to easily estimate the potentials from data as they might have a complex dependency to the rest of the graph. However, if the individual components are triangulated, the factorisation can be stated in terms of marginal probabilities over the variables in a clique.

Algorithm 1 learn CL parameters

```
Require: chain logic theory $T$, assumables $\mathcal{A}'$, observables $\mathcal{O}$, database $D$
  for $a \in \mathcal{A}'$ do
    $\texttt{Effect}(a) \leftarrow \{o \in \mathcal{O} \mid \exists\, H \in \text{CH} : T \cup H \models o \text{ and } T \cup (H \setminus \{a\}) \not\models o\}$
    $\texttt{Rel}(a) \leftarrow \{o \in \mathcal{O} \mid \forall H \in \text{CH}\colon a \in H \land T \cup H \models \texttt{Effect}(a) \text{ implies } T \cup H \models o\}$
  end for
  $V \leftarrow \mathcal{A}'$
  $E \leftarrow \{(a, a') \in \mathcal{A}' \times \mathcal{A}' \mid \texttt{Rel}(a) \cap \texttt{Rel}(a') \neq \varnothing\}$
  $JG \leftarrow (V, E)$ with separator set $\texttt{Rel}(a) \cap \texttt{Rel}(a')$ associated to every edge $(a, a')$
  let the weight of an edge in $JG$ be the cardinality of its separator set
  $J \leftarrow$ spanning tree of maximal weight of $JG$
  DJ $\leftarrow$ any directed tree of $J$
  for $a \in \mathcal{A}'$ do
    let $S$ be the union of separators of $a$ with its parents in DJ
    $\hat{\omega}(a) \leftarrow N_{\texttt{Effect}(a) \cup S} / N_S$
  end for
```

The proposed algorithm for determining the parameters is inspired by the use of a junction tree for probabilistic inference. The reason for this is that a junction tree provides sufficient information about the interactions between assumables, i.e., when they influence the same observables. Junction trees, with required properties such as the running intersection property, are only guaranteed to exist when the graph is triangulated, so we restrict ourselves to this case. Let $\mathcal{O}$ be the set of grounded non-assumables, which are the *observations.* As a convenience we write N_O with $O \subseteq \mathcal{O}$ for the number of tuples of D that contain O. In the following, we will assume that all $o \in \mathcal{O}$ are present in database D.

The learning procedure is described in Algorithm 1. It will first identify effects of assumables (`Effect`), which are those observables that are implied by the assumables, possibly together with other assumables. Then, indirect relation (`Rel`) are identified, which are those observables that are always included in the derivation to the effects from an assumable. The latter can be used to build up a junction tree, where variables are instantiated for a particular value. From this structure, weights of the assumables can be learned. The properties of junction trees ensure that joint distribution corresponds to the relative frequency of observables, i.e., we have the following, general, result.

Theorem 3. *Given a chain logic theory T with associated (moralised, triangulated) chain graph G and database D, then after running Algorithm 1, the resulting weight declaration described by $\hat{\omega}$ and T will be such that:*

$$P_T(O) = \frac{N_O}{N_\varnothing} \tag{9}$$

for all possible observations $O \subseteq \mathcal{O}$.

Proof (sketch). If the assumable a models the potential function of some clique, then the set $\texttt{Rel}(a)$ contains the nodes of that clique. Furthermore, JG is isomorph to a junction graph of the underlying chain graph G. According to [6, Theorem 1], then J will be a junction tree of G. Because of the equivalence between reasoning in chain logic and chain graphs, we have:

$$P(x_V) = \prod_C \prod_{M \in M(C)} \hat{\omega}(\varphi_M)$$

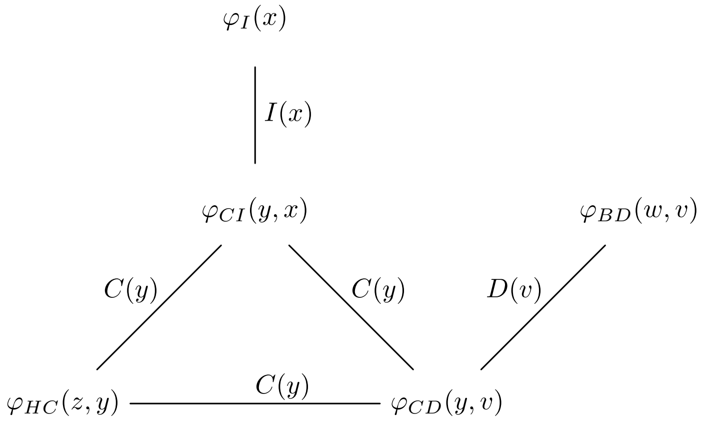
(see Eq. 7). Since $\hat{\omega}(\varphi_M) = N_M/N_S$ where S is some separator, it follows that

$$\prod_C \prod_{M \in M(C)} \hat{\omega}(\varphi_M) = \frac{\prod_C \prod_{M \in M(C)} N_M}{R}$$

where R amounts to the product of the frequency of separators on the edges. Then, observe that all separators of the graph are there iff they are in R as each separator appears exactly once in an edge of a junction tree.

Finally, by application of [7, Lemma 1] and [8, proposition 12.3.2], this frequency coincides exactly with the frequency interpretation of $P(X_V)$, i.e., coincides with the relative frequency of the data. □

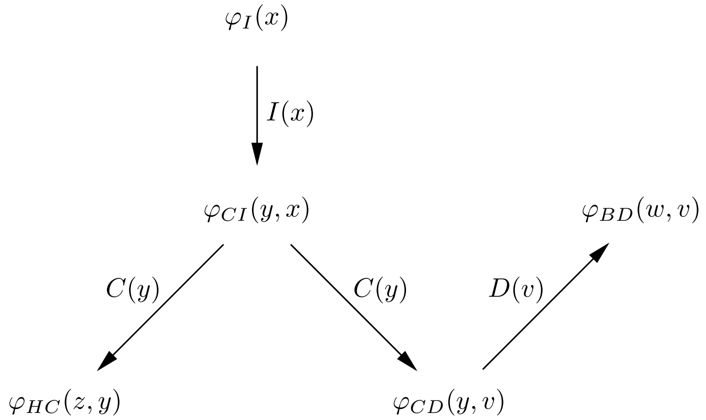
Example 5. Reconsider the chain graph of Fig. 2. By logical reasoning, it can be shown that: $\texttt{Rel}(\varphi_I(t)) = \{I(t)\}, \texttt{Rel}(\varphi_{CI}(t,t)) = \{C(t), I(t)\}$, etc, which gives the following graph (here shown with quantified variables to visualise the similarity to regular junction graphs):



A tree can be obtained from this graph by removing any of the edges from the loop in the undirected graph. In fact, it does not matter which one to choose as, if there is a loop, then the separators are all the same set of variables. Equation 9 implies that for chain logic it is irrelevant which one is chosen, but some are more closely related to the original graph as others. For example, if we take $\varphi_I(x)$ as a root, then $\varphi_{CI}(y,x)$ is a conditional probability, as in the original graph. If, on the other hand we take $\varphi_I(x)$ as a leaf, then its weight will be 1 and thus $\varphi_{CI}(y,x)$ will be the *joint* probability of C and I, given its parent.

In order to obtain the interpretation of the original graph, we adapt Algorithm 1 by choosing DJ as the maximum weight spanning tree such that there is an arc from a to a' iff $s(a, a') \in \texttt{Effect}(a)$ and $s(a, a') \notin \texttt{Effect}(a')$, where $s(a, a')$ denotes $\texttt{Rel}(a) \cap \texttt{Rel}(a')$, i.e., whenever an assumable a' does not explain an observable it is related to, which means it must be conditioned on this observation. For triangulated chain graphs, it can be proven that such a tree exists.

Example 6. In the example above, we thus take the tree with arrows between φ_I and φ_{CI}, and between φ_{CI} and φ_{CD}, giving, e.g., the following tree:



The learning algorithm will then, e.g., learn that:

$$\varphi_{CD}(x, y) = N_{\{C(x), D(y)\}} / N_{C(y)}$$

which corresponds to exactly the relative frequencies associated to variables in the original chain graph showing the relation between the two formalisms for learning.

Relational domains can be represented, and thus learned about using the same machinery, as long as the modeller ensures properties characterised by the class of triangulated chain graphs.

5 Comparison

Probabilistic Horn logic was originally proposed by Poole in [1]. It offers a framework that was shown to be as powerful as Bayesian networks, yet it has the advantage that it is a first-order language that integrates probabilistic and logical reasoning in a seamless fashion. Besides some changes in the terminology (such as using *weight* declarations in place of *disjoint* ones), the main differences in terms of syntax is the set of integrity constraints allowed and the probabilistic information captured in each formalism. Weights can sum up to any value, enabling the formalisation of potential functions instead of a (normalised) probability distribution. Furthermore, in our case, by allowing the use of extra integrity constraints, we are able to establish dependences among instantiations of hypotheses.

Those differences extend Poole's approach and allow us to obtain a more generic probabilistic model, being crucial for the representation of chain graph models. The graphical representation associated with a Bayesian network does not offer a unique way to represent the independence information, which makes the interpretation of Bayesian networks cumbersome. In contrast, an advantage of using chain graphs as underlying model is representing associations (e.g., *coughing* and *dyspnoea* in Example 1), which cannot be defined in Bayesian networks. In fact, chain graphs can capture the class of equivalent Bayesian networks. By using potential functions we can represent the quantitative influence between variables in a clique. The additional integrity constraints guarantee that instantiations of those potentials functions appear consistently in each explanation. Despite such differences, we still share with Poole's approaches some assumptions and similar results, for instance, with respect to the probability densities defined over the theory.

Bayesian logic programs [2] have similar limitations as probabilistic Horn logic; in addition, they are only proposed as formalisms to specify Bayesian networks in a logical way and reasoning is done in the generated Bayesian networks. Furthermore, the framework of Markov logic networks [3] has been proposed as a powerful language based on first-order logic to specify Markov networks. Yet, Markov networks are seen by researchers in probabilistic graphical models as the weakest type of such models, as much of the subtleties of representing conditional independence cannot be handled by Markov networks. In fact, formulae in Markov logic can only model associations between literals, whereas causal knowledge cannot be represented, for instance, between *coughing* and *hoarseness*. Furthermore, despite its expressive power, Markov logic is a generative language, i.e., specifications are translated into the corresponding graphical model on which reasoning is then performed in a standard fashion. The aim of the presented research was to design an expressive probabilistic logic that supports probabilistic reasoning and learning guided by the structure of the logic.

6 Final Considerations

In this paper we presented a simple, yet powerful, language for representing, reasoning with and learning generic chain graph models. Besides being able to incorporate both Bayesian and Markov network models as special cases, we maintain a strong relation between logical and probabilistic reasoning.

Our language still presents some restrictions. First, we use finite set of constants, which prohibits the use of continuous variables. For Markov logic networks, it has been shown that special cases of such networks can be extended to infinite domains by defining a Gibbs measure over sets of interpretations for logical formulae [9]. A similar approach could be taken here by defining a measure over the set of consistent states. Another limitation is the acyclicity assumption, which restricts the explicit representation of undirected graphs components. Even though we require certain assumptions for a sound probabilistic interpretation, weakening acyclicity seems feasible [10].

While we have shown in this paper that chain logic is powerful enough to define, reason, and learn about chain graphs, we have no strong reason to suspect that chain logic is restricted to this class of probabilistic graphical models. Although chain graphs form a fairly general class of probabilistic graphs, it might be the case that the language is applicable to a broader set of graphs. Also, modelling the independence implied in chain logic theories into a graphical model is an open question that will be investigated further.

With respect to learning, we have presented in this paper parameter learning of chain graph theories. Learning the structure of such graphs will be a subject of further research, but techniques from the inductive logic programming have been successful for learning Bayesian logic programs [11]. We believe similar ideas can be applied for learning chain logic theories.

Acknowledgements

The first and last author have been supported by the OCTOPUS project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Embedded Systems Institute program. The second author acknowledges the support through the B-Screen project funded by the Netherlands Organisation for Scientific Research (NWO) under BRICKS/FOCUS grant number 642.066.605.

References

1. Poole, D.: Probabilistic Horn abduction and Bayesian networks. AI Journal **64**(1) (1993) 81–129
2. Kersting, K., de Raedt, L.: Bayesian logic programs. Technical Report 151, Institute for Computer Science - University of Freiburg (2001) CoRR cs.AI/0111058.
3. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning **62** (2006) 107–136
4. Pearl, J.: Probabilistic Reasoning in Inteligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
5. Lauritzen, S.L.: Graphical Models. Oxford:Clarendon (1996)
6. Jensen, F., Jensen, F.: Optimal junction trees. In: Uncertainty in Artificial Intelligence. (1994)
7. Bouckaert, R., Studený, M.: Chain graphs: Semantics and expressiveness. In: Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Volume 946 of LNCS., Springer (1995) 69–76
8. Whittaker, J.: Graphical Models in Applied Multivariate Statistics. Wiley (1990)
9. Singla, P., Domingos, P.: Markov logic in infinite domains. In: Proc. of UAI'07, AUAU Press (2007) 368–375
10. Poole, D.: The independent choice logic for modelling multiple agents under uncertainty. AI Journal **94**(1–2) (1997) 7–56
11. Kersting, K., de Raedt, L.: Towards combining inductive logic programming with bayesian networks. In Rouveirol, C., Sebag, M., eds.: Proc. ILP 2001. Volume 2157 of Lecture Notes in Artificial Intelligence., Springer (2001) 118–131