Markov Conditions and Factorization in Logical Credal Networks

Fabio Gagliardi Cozman
Universidade de São Paulo, Brazil

Abstract

We examine the recently proposed language of *Logical Credal Networks*, in particular investigating the consequences of various Markov conditions. We introduce the notion of structure for a Logical Credal Network and show that a structure without directed cycles leads to a well-known factorization result. For networks with directed cycles, we analyze the differences between Markov conditions, factorization results, and specification requirements.

Keywords: logical credal networks, probabilistic logic, Markov condition, factorization.

1. Introduction

This paper examines *Logical Credal Networks*, a formalism recently introduced by Marinescu et al. [14] to combine logical sentences, probabilities and independence relations. They have proposed interesting ideas and evaluated the formalism in practical scenarios with positive results.

The central element of a Logical Credal Network (LCN) is a collection of constraints over probabilities. Independence relations are then extracted mostly from the logical content of those inequalities. This scheme differs from previous proposals that extract independence relations from explicitly specified graphs [1, 7, 8]. Several probabilistic logics have also adopted explicit syntax for independence relations even when graphs are not employed [2, 9, 10].

While Logical Credal Networks have points in common with existing formalisms, they do have novel features that deserve attention. For one thing, they resort to directed graphs that may contain directed cycles. Also they are endowed with a sensible Markov condition that is distinct from previous ones. Little is known about the consequences of these features, and how they interact with the syntactic conventions that turn logical formulas into edges in graphs. In particular, it seems that no study has focused on the consequences of Markov conditions on factorization results; that is, how such conditions affect the factors that constitute probability distributions.

In this paper we seek a deeper understanding of Logical Credal Networks, looking at their specification, their Markov conditions, their factorization properties. We introduce the notion of "structure" for a LCN. We then show that the local Markov condition proposed by Marinescu et al. [14] collapses to the usual local Markov condition applied to chain graphs when the structure has no directed cycles. We analyze the behavior of the former Markov condition in the presence of directed cycles, in particular investigating factorization properties and discussing the semantics of the resulting language. We also suggest novel semantics for LCNs and examine related factorization results.

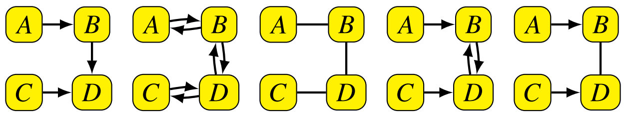
2. Graphs and Markov Conditions

In this section we present the necessary concepts related to graphs and graph-theoretical probabilistic models (Bayesian networks, Markov networks, and chain graphs). Definitions and notation vary across the huge literature on these topics; we rely here on three sources. We use definitions by Marinescu et al. [14] and by Spirtes [18] in their work on LCNs and on directed graphs respectively; we also use standard results from the textbook by Cowell et al. [5].

A *graph* is a triple $(\mathcal{V}, \mathcal{E}_D, \mathcal{E}_U)$, where $\mathcal{V}$ is a set of nodes, and both $\mathcal{E}_D$ and $\mathcal{E}_U$ are sets of edges. A node is always labeled with the name of a random variable; in fact, we do not distinguish between a node and the corresponding random variable. The elements of $\mathcal{E}_D$ are *directed* edges. A directed edge is an ordered pair of distinct nodes, denoted by $A \rightarrow B$. The elements of $\mathcal{E}_U$ are *undirected* edges. An undirected edge is a pair of distinct nodes, denoted by $A \sim B$; note that nodes are not ordered in an undirected edge, so there is no difference between $A \sim B$ and $B \sim A$. Note that $\mathcal{E}_D$ and $\mathcal{E}_U$ are sets, so there are no multiple copies of elements in them (for instance, there are no multiple undirected edges between two nodes). Note also that there is no loop from a node to itself.

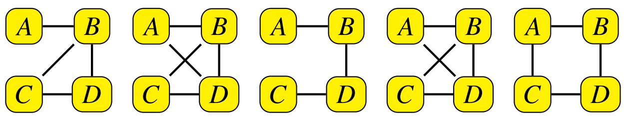
If there is a directed edge from A to B, the edge is said to be *from* A to B, and then A is a *parent* of B and B is a *child* of A. The parents of A are denoted by $\mathrm{pa}(A)$. If there are directed edges $A \rightarrow B$ and $B \rightarrow A$ between A and B, we say there is *bi-directed* edge between A and B and write $A \rightleftarrows B$. If $A \sim B$, then both nodes are said to be *neighbors*. The neighbors of A are denoted by $\mathrm{ne}(A)$. The *boundary* of a node A, denoted by $\mathrm{bd}(A)$, is the set $\mathrm{pa}(A) \cup \mathrm{ne}(A)$. The boundary of a set $\mathcal{B}$ of nodes is $\mathrm{bd}(\mathcal{B}) = \cup_{A \in \mathcal{B}} \mathrm{bd}(A) \backslash \mathcal{B}$. If we have a set $\mathcal{B}$ of nodes such that, for all $A \in \mathcal{B}$, the boundary of A is contained in $\mathcal{B}$, then $\mathcal{B}$ is an *ancestral set*.

© 2023 F.G. Cozman.



(a) (b) (c) (d) (e)

Figure 1: Graphs (directed/directed/undirected/directed/chain). We have $\text{pa}(B) = \{A\}$ in Figures 1.a and 1.e, $\text{pa}(B) = \{A, D\}$ in Figures 1.b and 1.d, and $\text{pa}(B) = \emptyset$ in Figure 1.c.



(a) (b) (c) (d) (e)

Figure 2: The moral graphs of the graphs in Figure 1.

A *path* from A to B is a sequence of distinct edges, the first one between A and some node C_1, then between C_1 and C_2 and so on, until an edge between C_k and B, where all nodes are distinct with the exception that the first and last nodes may be identical, and such that for each pair (D_1, D_2) of consecutive nodes in the path we have either $D_1 \rightarrow D_2$ or $D_1 \sim D_2$ but never $D_2 \rightarrow D_1$. If A and B are in fact identical, the path is a *cycle*. If there is at least one directed edge in a path, the path is a *directed path*; if that path is a cycle, then it is a *directed cycle*. If a path is not directed, then it is *undirected* (hence all edges in the path are undirected ones). A *directed/undirected* graph is a graph that only contains directed/undirected edges. A graph without directed cycles is a *chain graph* (note that such a graph may have a cycle consisting only of undirected edges). Figure 1 depicts a number of graphs.

If there is a directed path from A to B, then A is an *ancestor* of B and B is a *descendant* of A. For instance, in Figure 1.a, A is the ancestor of B and D is the descendant of B; in Figure 1.c, there are no ancestors nor descendants of B; in Figure 1.e, A is the ancestor of B, and there are no descendants of B. As a digression, note that Cowell et al. [5] define "ancestor" and "descendant" somewhat differently, by asking that there is a path from A to B but not from B to A; this definition is equivalent to the previous one for graphs without directed cycles, but it is different otherwise (for instance, in Figure 1.b the node B has descendants $\{A, C, D\}$ in the previous definition but no descendant in the sense of Cowell et al. [5]). We stick to our former definition, a popular one [13] that seems appropriate in the presence of directed cycles [18].

We will need the following concepts:

- Suppose we take graph $\mathcal{G}$ and remove its directed edges to obtain an auxiliary undirected graph $\mathcal{G}'$. A set of nodes $\mathcal{B}$ is a *chain multi-component* of $\mathcal{G}$ iff every pair of nodes in $\mathcal{B}$ is connected by a path in $\mathcal{G}'$. And $\mathcal{B}$ is a *chain component* iff it is *either* a chain multi-component *or* a single node that does not belong to any chain multi-component.

- Suppose we take graph $\mathcal{G}$ and add undirected edges between each two nodes nodes that have a children in a common chain component of $\mathcal{G}$ and that are not already joined in $\mathcal{G}$. Suppose we then take the resulting graph and transform every directed edge into an undirected edge (if $A \leftrightarrows B$, then both transformed directed edges collapse into $A \sim B$). The final result is the *moral graph* of $\mathcal{G}$, denoted by $\mathcal{G}^m$.

- Suppose we take a graph $\mathcal{G}$ and a triple $(\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3)$ of disjoint subsets of nodes, and we build the moral graph of the smallest ancestral set containing the nodes in $\mathcal{N}_1 \cup \mathcal{N}_2 \cup \mathcal{N}_3$. The resulting graph is denoted by $\mathcal{G}^{ma}(\mathcal{N}_1 \cup \mathcal{N}_2 \cup \mathcal{N}_3)$.

Figure 2 depicts the moral graphs that correspond respectively to the five graphs in Figure 1.

There are several formalisms that employ graphs to represent stochastic independence (and dependence) relations among the random variables associated with nodes. In this paper we focus only on discrete random variables, so the concept of stochastic independence is quite simple: random variables X and Y are (conditionally) independent given random variables Z iff $\mathbb{P}(X = x, Y = y|Z = z) = \mathbb{P}(X = x|Z = z)\,\mathbb{P}(Y = y|Z = z)$ for every possible x and y and every z such that $\mathbb{P}(Z = z) > 0$. In case Z is absent, we have independence of X and Y iff $\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x)\,\mathbb{P}(Y = y)$ for every possible x and y.

A *Markov condition* explains how to extract independent relations from a graph; there are many such conditions in the literature [5].

Consider first an undirected graph $\mathcal{G}$ with set of nodes $\mathcal{N}$. The *local Markov condition* states that a node A is independent of all nodes in $\mathcal{N}$ other than A itself and A's neighbors, $\text{ne}(A)$, given $\text{ne}(A)$. The *global Markov condition* states that, given any triple $(\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3)$ of disjoint subsets of $\mathcal{N}$, such that $\mathcal{N}_2$ separates $\mathcal{N}_1$ and $\mathcal{N}_3$, then nodes $\mathcal{N}_1$ and $\mathcal{N}_3$ are independent given nodes $\mathcal{N}_2$.[1] If a probability distribution over all random variables in $\mathcal{G}$ is everywhere larger than zero, then both conditions are equivalent and they are equivalent to a *factorization property*: for each configuration of variables $X = x$, where X denotes the random variables in $\mathcal{G}$, we have $\mathbb{P}(X = x) = \prod_{c \in C} \phi_c(x_c)$, where C is the set of cliques of $\mathcal{G}$, each ϕ_c is a function over the random

[1] In an undirected graph, a set of nodes separates two other sets iff, by deleting the separating nodes, we have no connecting path between a node in one set and a node in the other set.

variables in clique c, and x_c is the projection of x over the random variables in clique c.[2]

Now consider an acyclic directed graph $\mathcal{G}$ with set of nodes $\mathcal{N}$. The *local Markov condition* states that a node A is independent, given A's parents $\mathrm{pa}(A)$, of all its non-descendants non-parents except A itself. The factorization produced by the local Markov condition is

$$\mathbb{P}(X = x) = \prod_{N \in \mathcal{N}} \mathbb{P}\left(N = x_N \,|\mathrm{pa}(N) = x_{\mathrm{pa}(N)}\right), \quad (1)$$

where x_N is the value, in x, of random variable N and $x_{\mathrm{pa}(N)}$ is the projection of x over the parents of N.

Finally, consider a chain graph $\mathcal{G}$ with set of nodes $\mathcal{N}$. The *local Markov condition* for chain graphs is:

Definition 1 (LMC(C)) *A node A is independent, given its boundary, of all nodes that are not A itself nor descendants nor boundary nodes of A.*

The *global Markov condition* for chain graphs is significantly more complicated:

Definition 2 (GMC(C)) *Given any triple $(\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3)$ of disjoint subsets of $\mathcal{N}$, if $\mathcal{N}_2$ separates $\mathcal{N}_1$ and $\mathcal{N}_3$ in the graph $\mathcal{G}^{ma}(\mathcal{N}_1 \cup \mathcal{N}_2 \cup \mathcal{N}_3)$, then nodes $\mathcal{N}_1$ and $\mathcal{N}_3$ are independent given nodes $\mathcal{N}_2$.*

If a probability distribution over all random variables in $\mathcal{G}$ is everywhere larger than zero, then both Markov conditions are equivalent and they are equivalent to a *factorization property* as follows. Take the chain components $T_1, \dots, T_n$ ordered so that nodes in T_i can only be at the end of directed edges starting from chain components before T_i; this is always possible in a chain graph. Then the factorization has the form $\mathbb{P}(X = x) = \prod_{i=1}^{n} \mathbb{P}\left(\mathcal{N}_i = x_{\mathcal{N}_i} \mid \mathrm{bd}(\mathcal{N}_i) = x_{\mathrm{bd}(\mathcal{N}_i)}\right)$ where $\mathcal{N}_i$ is the set of nodes in the ith chain component; $x_{\mathcal{N}_i}$ and $x_{\mathrm{bd}(\mathcal{N}_i)}$ are respectively the projection of x over $\mathcal{N}_i$ and $\mathrm{bd}(\mathcal{N}_i)$. Moreover, each factor in the product itself factorizes accordingly to an undirected graph that depends on the corresponding chain component [5]. More precisely, for each chain component T_i, build an undirected graph consisting of the nodes in $\mathcal{N}_i$ and $\mathrm{bd}(\mathcal{N}_i)$ with all edges between these nodes in $\mathcal{G}$ turned into undirected edges in this new graph, and with new undirected edges connecting each pair of nodes in $\mathrm{bd}(\mathcal{N}_i)$ that were not joined already; then each $\mathbb{P}(\mathcal{N}_i \mid \mathrm{bd}(\mathcal{N}_i))$ equals the ratio $\phi_i(\mathcal{N}_i, \mathrm{bd}(\mathcal{N}_i))/\phi_i(\mathrm{bd}(\mathcal{N}_i)))$ for positive function ϕ_i, where $\phi_i(\mathrm{bd}(\mathcal{N}_i))) = \sum \phi_i(\mathcal{N}_i, \mathrm{bd}(\mathcal{N}_i))$ with the sum extending over all configurations of $\mathcal{N}_i$.

[2]A clique is a maximal set of nodes such that each pair of nodes in the set is joined.

3. Logical Credal Networks

A Logical Credal Network (LCN) consists of a set of propositions $\mathcal{N}$ and two sets of constraints $\mathcal{T}_U$ and $\mathcal{T}_D$. The set $\mathcal{N}$ is finite with propositions $A_1, \dots, A_n$. Each proposition A_i is associated with a random variable X_i that is an indicator variable: if A_i holds in an interpretation of the propositions then $X_i = 1$; otherwise, $X_i = 0$. From now on we simply use the same symbol for a proposition and its corresponding indicator random variable. Each constraint in $\mathcal{T}_U$ and in $\mathcal{T}_D$ is of the form

$$\alpha \leq \mathbb{P}(\phi|\varphi) \leq \beta,$$

where each ϕ and each φ is a formula. In this paper, formulas are propositional (with propositions in $\mathcal{N}$ and connectives such as negation, disjunction, conjunction). The definition of LCNs by Marinescu et al. [14] allows for relational structures and first-order formulas; however, their semantics is obtained by grounding on finite domains, so we can focus on a propositional language for our purposes here.

Note that φ can be a tautology $\top$, in which case we can just write the "unconditional" probability $\mathbb{P}(\phi)$. One can obviously use simple variants of constraints, such as $\mathbb{P}(\phi|\varphi) = \beta$ or $\mathbb{P}(\phi|\varphi) \geq \alpha$ or $\mathbb{P}(\phi) \leq \alpha$, whenever needed.

The semantics of a LCN is given by a translation from the LCN to a directed graph where each proposition/random variable is a node (we often refer to them as *proposition-nodes*). Each constraint is then processed as follows. First, a node labeled with formula ϕ is added and, in case φ is not $\top$, another node labeled with φ is added (we often refer to them as *formula-nodes*), with a directed edge from φ to ϕ. Then an edge is added from each proposition in φ to node φ in case the latter is in the graph, and an edge is added from node ϕ to each proposition in ϕ.[3] Finally, in case the constraint is in $\mathcal{T}_U$, an edge is added from each proposition in ϕ to node ϕ. We do not distinguish between two logically equivalent formulas (the original proposal by Marinescu et al. [14] focused only on syntactic operations).

The graph just described is referred to as the *primal graph* of the LCN. As shown in the next example, in our drawings formulas appear inside dashed rectangles. For the sake of simplicity, we remove such a rectangle whenever the corresponding formula contains a single proposition and its connections to surrounding nodes can be inferred from context; in such cases we can just connect edges from and to the corresponding proposition-nodes.

Example 1 *Consider the following LCN, based on the Smokers and Friends example by Marinescu et al. [14]. We*

[3]We note that the original presentation of LCNs is a bit different from what we just described, as there are no edges added for a constraint in $\mathcal{T}_D$ for which φ is $\top$. But this does not make any difference in the results and simplifies a bit the discussion.

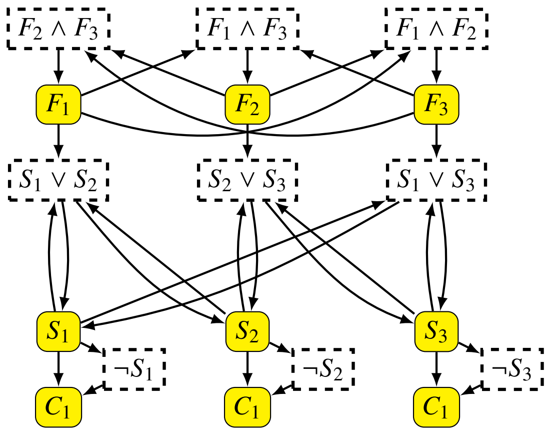


Figure 3: The primal graph of the LCN in Example 1.

have propositions C_i, F_i, S_i for $i \in \{1,2,3\}$. All constraints belong to $\mathcal{T}_U$ (that is, $\mathcal{T}_D$ is empty), with $i, j \in \{1,2,3\}$:

$$
\begin{array}{ll}
0.5 \leq \mathbb{P}\big(F_i | F_j \wedge F_k\big) \leq 1, & i \neq j, i \neq k, j \neq k; \\
0 \leq \mathbb{P}\big(S_i \vee S_j | F_i\big) \leq 0.2, & i \neq j; \\
0.03 \leq \mathbb{P}(C_i | S_i) \leq 0.04; & \\
0 \leq \mathbb{P}(C_i | \neg S_i) \leq 0.01. &
\end{array}
$$

The primal graph of this LCN is depicted in Figure 3. Note that there are several directed cycles in this primal graph.

Marinescu et al. [14] then define:

Definition 3 *A* lcn-parent *of a proposition A is a proposition such that there exists a directed path in the primal graph from it to A in which all intermediate nodes are formulas.*

The set of lcn-parents of A is denoted by lcn-pa(A).

Definition 4 *A* lcn-descendant *of a proposition A is a proposition such that there exists a directed path in the primal graph from A to it in which no intermediate node is a lcn-parent of A.*

The set of lcn-descendants of A is denoted by lcn-de(A).

The connections between these concepts and the definitions of parent and descendant in Section 2 will be clear in the next section.

In any case, using these definitions Marinescu et al. [14] propose a Markov condition:

Definition 5 (LMC(LCN)) *A node A is independent, given its lcn-parents, of all nodes that are not A itself nor lcn-descendants of A nor lcn-parents of A.*

The Markov condition in Definition 5 is:

$$X \perp\!\!\!\perp \mathcal{N} \backslash \{\{A\} \cup \text{lcn-de}(A) \cup \text{lcn-pa}(A)\} \mid \text{lcn-pa}(A), \quad (2)$$

where we use $\perp\!\!\!\perp$ here, and in the remainder of the paper, to mean "is independent of".

We will often use the superscript c to mean complement, hence $\mathcal{A}^c \doteq \mathcal{N} \backslash \mathcal{A}$.

Marinescu et al. [14] have derived inference algorithms (that is, they consider the computation of conditional probabilities) that exploit such independence relations, and they examine applications that demonstrate the practical value of LCNs.

It seems that a bit more discussion about the meaning of this Markov condition, as well as its properties and consequences, would be welcome. To do so, we find it useful to introduce a novel concept, namely, the *structure* of a LCN.

4. The Structure of a LCN

The primal graph of a LCN is rather similar in spirit to the *factor graph* of a Bayesian network [13], where both random variables and conditional probabilities are explicitly represented. This is a convenient device when it comes to message-passing inference algorithms, but perhaps it contains too much information when one wishes to examine independence relations.

We introduce another graph to be extracted from the primal graph of a given LCN, that we call the *structure* of the LCN, as follows:

1. For each formula-node ϕ that appears as a conditioned formula in a constraint in $\mathcal{T}_U$, place an undirected edge between any two propositions that appear in ϕ.

2. For each pair of formula-nodes φ and ϕ that appear in a constraint, add a directed edge from each proposition in φ to each proposition in ϕ.

3. If, for some pair of proposition-nodes A and B, there is now a pair of edges $A \leftrightarrows B$, then replace both edges by an undirected edge.

4. Finally, remove the formula-nodes and all edges in and out of them; and if there are identical edges left, remove duplicates.

Example 2 *Figures 4.a and 4.b depict the structure of the LCN in Example 1.*

We have:

Lemma 6 *The set of lcn-parents of a proposition A in a LCN is identical to the boundary of A with respect to the structure of the LCN.*

Proof Consider a LCN with a primal graph $\mathcal{D}$. If B is a lcn-parent of A with respect to $\mathcal{D}$, then constructs such as $B \to \varphi \to \phi \to A$ or $B \leftrightarrows \phi \leftrightarrows A$ must be present, leading

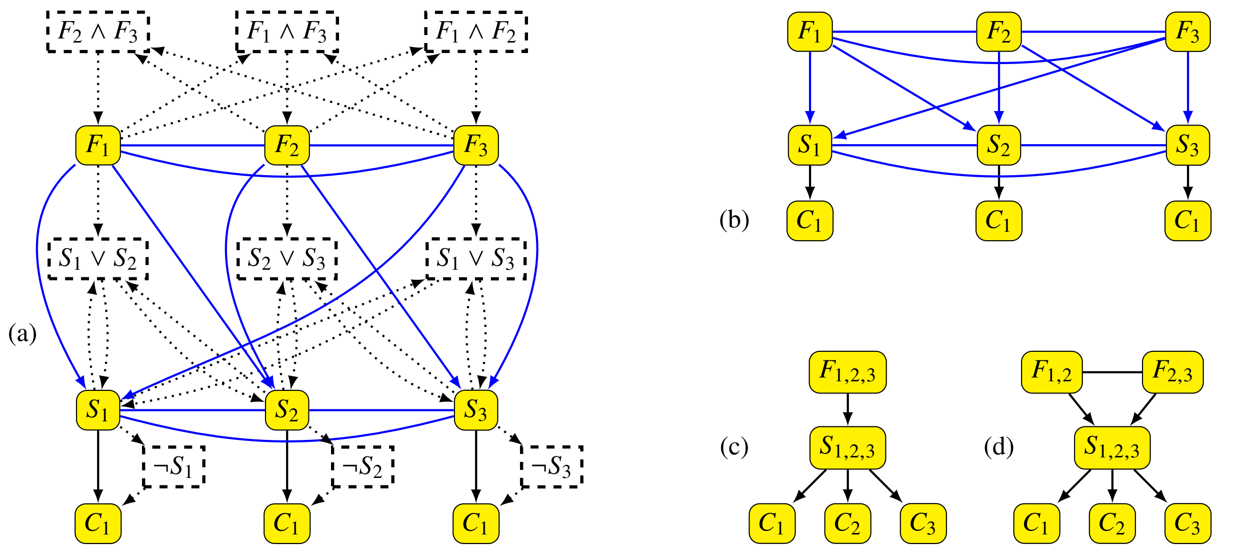


Figure 4: (a) The primal graph for the LCN in Example 1, together with the edges in the structure of the LCN. Edges in the structure are solid (the ones added in the process are in blue); edges in and out of formula-nodes are dotted. (b) The structure of the LCN, by removing the formula-nodes and associated edges. (c) A directed acyclic graph with the chain components of the chain graph that represents the structure. (d) A variant discussed in Example 3.

either to direct or undirected edges in the structure; hence B appears either as a parent of A or a neighbor in the structure of the LCN. Conversely, if B is a parent or a neighbor of A in the structure of the LCN, then one of the sequences of edges already mentioned must be in $\mathcal{D}$, so B is a lcn-parent of A in $\mathcal{D}$. ■

The natural candidate for the concept of "descendant" in a structure, so as to mirror the concept of lcn-descendant, is as follows:

Definition 7 *If there is a directed path from A to B such that no intermediate node is a boundary node of A, then B is a* strict descendant *of A.*

Using the previous definitions, we can state a local Markov condition that works for any graph:

Definition 8 (LMC(C-STR)) *A node A is independent, given its boundary, of all nodes that are not A itself nor strict descendants of A nor boundary nodes of A.*

In symbols,

$$X \perp\!\!\!\perp \mathcal{N} \backslash \{\{A\} \cup \mathrm{sde}(A) \cup \mathrm{bd}(A)\} \mid \mathrm{bd}(A), \qquad (3)$$

where $\mathrm{sde}(A)$ denotes the set of strict descendants of A.

It is not immediately clear that LMC(LCN) and LMC(C-STR) are equivalent when applied to a structure, because the lcn-descendants of a node in a primal graph are not the strict descendants of the same node in the corresponding structure. To see this, consider Figure 1.b: if we think of this graph as containing the proposition nodes and their connections from the primal graph, then A and D are the lcn-descendants of B; in the corresponding structure, given by Figure 1.c, B has no strict descendants as there are no directed paths.

In fact, the set of lcn-descendants of a proposition node A, lcn-de(A), can be divided in two sets of nodes, lcn-de$_1(A)$ and lcn-de$_2(A)$. Take a node B to be in lcn-de$_1(A)$ if and only if there is at least one directed path from A to B that starts with a directed edge and such that no intermediate node is a lcn-parent of A. All of those nodes are strict descendants of A because the directed paths that exist in the primal graph are translated verbatim to directed paths in the corresponding structure. And take a node B to be in lcn-de$_2(A)$ if and only if all directed paths from A to B that have no lcn-parent of A as an intermediate node start with a bi-directed edge. By construction, lcn-de$_1(A)$ and lcn-de$_2(A)$ are disjoint. Now note that any node B in lcn-de$_2(A)$ must be a lcn-parent of A (if there is B that is not a lcn-parent of A, we have a situation where all paths from A to B go through a lcn-parent of A, and by definition this is not possible). Note also that nodes in lcn-de$_2(A)$ appear as neighbors of A in the corresponding structure of the LCN.

Consequently, the nodes that are lcn-descendants of A are *either* strict descendants of A *or* neighbors of A with respect to the structure.

Using these insights, we can show that the LMC(LCN) and the LMC(C-STR) do have the same effect when applied to the structure:

Theorem 9 *Given a LCN, the Markov condition LMC(LCN) in Definition 5 is identical, with respect to the independence relations it imposes, to the local Markov condition LMC(C-STR) in Definition 8 applied to the structure of the LCN.*

Proof To prove that Expressions (2) and (3) are equivalent, consider first the following equalities with respect to sets of nodes:

$$\begin{aligned} \text{lcn-de}(A) \cup \text{lcn-pa}(A) &= \text{lcn-de}_1(A) \cup \text{lcn-de}_2(A) \\ &\quad \cup \text{lcn-pa}(A) \\ &= \text{sde}(A) \cup \text{ne}(A) \\ &\quad \cup \text{pa}(A) \cup \text{ne}(A) \\ &= \text{sde}(A) \cup \text{pa}(A) \cup \text{ne}(A) \\ &= \text{sde}(A) \cup \text{bd}(A), \end{aligned}$$

and consequently $\{A\} \cup \text{lcn-de}(A) \cup \text{lcn-pa}(A)$ is equal to $\{A\} \cup \text{sde}(A) \cup \text{bd}(A)$, hence

$$\{\{A\} \cup \text{lcn-de}(A) \cup \text{lcn-pa}(A)\}^c = \{\{A\} \cup \text{sde}(A) \cup \text{bd}(A)\}^c.$$

Now note that $\text{lcn-pa}(A) = \text{bd}(A)$, by Lemma 6, to obtain the desired result. ■

Note that, in the presence of directed cycles, we may have lcn-parents that are lcn-descendants, and similarly we may have parents and neighbors that are strict descendants. The only real difference between lcn-descendants and strict descendants is this: for a given node A, there may be a node B that is a neighbor of A and that is not reached through any directed cycle emanating from A; this node B is a lcn-descendant of A, but not a strict descendant of A. This difference however does not lead to a difference between LMC(LCN) and LMC(C-STR), as shown by Theorem 9.

5. Chain Graphs and Factorization

If the structure of a LCN is a directed acyclic graph, the LMC(LCN) mimics the usual local Markov conditions for Bayesian networks (also adopted in credal networks [6, 15]). If instead all constraints in a LCN belong to $\mathcal{T}_U$ and all of them only refer to "unconditional" probabilities (that is, $\varphi = \top$ in every constraint), then the structure of the LCN is an undirected graph endowed with the usual local Markov condition for undirected graphs.

These previous points can be generalized in a satisfying way *whenever the structure contains no directed cycle*:

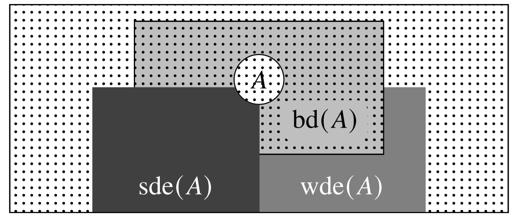


Figure 5: Proposition nodes of a LCN whose structure is a chain graph. The dotted area contains the nodes that are not descendants. In a chain graph, the strict descendants cannot be in the boundary.

Theorem 10 *If the structure of a LCN is a chain graph, and probabilities are positive, then the Markov condition LMC(LCN) in Definition 5 is identical, with respect to the independence relations it imposes, to the LMC(C) applied to the structure.*

Before we prove this theorem, it should be noted that sets of descendants and of strict descendants are not identical. This is easy to see in graphs with directed cycles: in Figure 1.b, node B has descendants $\{A, C, D\}$ and strict descendants $\{A, D\}$. But even in chain graphs we may have differences: for instance, suppose that in Figure 1.e we add a single directed edge from D to a new node E; then E is the only descendant of B, but B has no strict descendants.

In fact, the descendants of a node A can be divided into two sets, much as we did in connection with the lcn-descendants of a node. A node B is in the first set if and only if there is at least one directed path from A to B that starts with a directed edge. All of those nodes are strict descendants of A. A node C is in the second set if and only if all directed paths from A to C start with an undirected edge. Then all directed paths from A to C first reach a node that is in the boundary of A and consequently C is *not* a strict descendant of A. The first set is thus exactly $\text{sde}(A)$, and we migh refer to the second set as $\text{wde}(A)$, the set of "*weak*" descendants of A. By construction we have $\text{sde}(A) \cap \text{wde}(A) = \emptyset$. Moreover, we conclude (see Figure 5) that

$$(\text{sde}(A))^c = (\text{de}(A))^c \ \cup \ \text{wde}(A),$$

from which we have that $\mathcal{N} \backslash (\{A\} \cup \text{sde}(A) \cup \text{bd}(A)) = \{A\}^c \cap ((\text{de}(A))^c \ \cup \ \text{wde}(A)) \cap (\text{bd}(A))^c$. Note that $(\{A\}^c \cap \text{wde}(A) \cap (\text{bd}(A))^c) = \text{wde}(A)$ by construction, hence $\mathcal{N} \backslash (\{A\} \cup \text{sde}(A) \cup \text{bd}(A))$ is the union of two disjoint sets:

$$(\{A\}^c \cap (\text{de}(A))^c \cap (\text{bd}(A))^c) \ \cup \ \text{wde}(A). \quad (4)$$

Proof Suppose we have the LMC(C-STR); so, for any node A, we have Expression (3). Using Expression (4), we have $A \perp\!\!\!\perp ((\{A\}^c \cap (\mathrm{de}(A))^c \cap (\mathrm{bd}(A))^c) \cup \mathrm{wde}(A)) | \mathrm{bd}(A)$; using the Decomposition property of probabilities,[4] we obtain $A \perp\!\!\!\perp \{A\}^c \cap (\mathrm{de}(A))^c \cap (\mathrm{bd}(A))^c | \mathrm{bd}(A)$. Thus the LMC(C) holds.

Suppose the LMC(C) holds; given the positivity condition, the GMC(C) holds [5]. Take a node A and suppose there is a node $B \in \mathcal{N} \backslash (\{A\} \cup \mathrm{sde}(A) \cup \mathrm{bd}(A))$. We now prove that B is separated from A by $\mathrm{bd}(A)$ in

$$\begin{aligned} \mathcal{G}^*_A &\doteq \mathcal{G}^{ma}(\{A\} \cup (\{A\}^c \cap (\mathrm{sde}(A))^c \cap (\mathrm{bd}(A))^c) \cup \mathrm{bd}(A)) \\ &= \mathcal{G}^{ma}((\mathrm{sde}(A))^c), \end{aligned}$$

and consequently the GMC(C) leads to $A \perp\!\!\!\perp (\{A\}^c \cap (\mathrm{sde}(A))^c \cap (\mathrm{bd}(A))^c) | \mathrm{bd}(A)$ as desired. Note that, by construction, nodes in $\mathrm{sde}(A)$ cannot be in the ancestral set of any node in $\mathrm{wde}(A)$, so $\mathrm{sde}(A)$ is not in $\mathcal{G}^*_A$. Consequently, $\mathcal{G}^*_A$ is the moral graph of the graph consisting of nodes in $(\mathrm{sde}(A))^c$ and edges among them in the structure. In the moral graph, paths from A to B may start by moving from A to a parent or a neighbor of A in the original structure of the LCN, and those paths are blocked by $\mathrm{bd}(A)$. The only other way to reach B from A in $\mathcal{G}^*_A$ would be to follow a path $\mathcal{G}^*_A$ starting with an edge connecting A and some node C such that A and C are both parents of a common chain component; however this cannot happen as all direct children of A are in $\mathrm{sde}(A)$, and $\mathcal{G}^*_A$ does not contain nodes in $\mathrm{sde}(A)$. Hence we must have separation of A and B by $\mathrm{bd}(A)$ in $\mathcal{G}^*_A$, and therefore we have separation of A and $\mathcal{N} \backslash (\{A\} \cup \mathrm{sde}(A) \cup \mathrm{bd}(A))$ by $\mathrm{bd}(A)$ as desired. ■

The significance of the previous theorem is that, assuming that all probabilities are positive, the local Markov condition for a chain graph is equivalent both to the global Markov condition and to the factorization property of chain graphs. This allows us to break down the probability distribution over all random variables in a LCN in hopefully much smaller pieces that require less specification effort.

Example 3 *Figure 4.b depicts a structure that is in fact a chain graph. We can group variables to obtain chain components $F_{1,2,3}$ and $S_{1,2,3}$ and draw a directed acyclic graph with the chain components, as in Figure 4.c. The joint probability distribution factorizes as Expression (1):*

$$\begin{aligned} \mathbb{P}(F_{1,2,3} = f, S_{1,2,3} = s, C_1 = c_1, C_2 = c_2, C_3 = c_3) &= \\ \mathbb{P}(F_{1,2,3} = f)\, \mathbb{P}(S_{1,2,3} = s | F_{1,2,3} = f) & \\ \mathbb{P}(C_1 = c_1 | S_{1,2,3} = s)\, \mathbb{P}(C_2 = c_2 | S_{1,2,3} = s) & \\ \mathbb{P}(C_3 = c_3 | S_{1,2,3} = s)\,, & \end{aligned}$$

where f is a configuration of the random variables in $F_{1,2,3}$, while s is a configuration of the random variables in $S_{1,2,3}$. Because there are no independence relations "inside" the chain components, this factorization is guaranteed even if some probability values are equal to zero [5].

Suppose that the three constraints $0.5 \leq \mathbb{P}(F_i | F_j \wedge F_k) \leq 1$ are replaced so that, instead of a single chain component with F_1, F_2, F_3, we have two chain components, one with F_1 and F_2, the other with F_2 and F_3. The chain components might be organized as in Figure 4.d. If all probabilities are positive, that chain graph leads to a factorization of the joint probability distribution similar to the previous one, but now $\mathbb{P}(F_{1,2,3} = f_1 f_2 f_3) = \phi_1(F_{1,2} = f_1 f_2)\phi_2(F_{2,3} = f_2 f_3)$, where ϕ_1 and ϕ_2 are positive functions, and the values of F_1, F_2 and F_3 are indicated by f_1, f_2, f_3 respectively.

The assumption that probabilities are positive is important: when some probabilities are zero, there is no guarantee that a factorization actually exists [16]. This is unfortunate as a factorization leads to valuable computational simplifications. One strategy then is to guarantee that all configurations do have positive probability, possibly by adding language directives that bound probabilities from below. This solution may be inconvenient if we do have some hard constraints in the domain. For instance, one may wish to impose that $A \vee B$ (in which case $\mathbb{P}(\neg A \wedge \neg B) = 0$).

However, is is still possible to obtain a factorization even when hard constraints are imposed. Say we have a formula, for instance $A \vee B$, that must be satisfied. We treat it as a constraint $1 \leq \mathbb{P}(A \vee B) \leq 1$ in $\mathcal{T}_U$, thus guaranteeing that there is a clique containing its propositions/random variables. Then we remove the impossible configurations of these random variables (in our running example, we remove $A = B = 0$), thus reducing the number of possible configurations for the corresponding clique. A factorization is obtained again in the reduced space of configurations, provided the remaining configurations do have positive probabilities.

Finally, an entirely different strategy may be pursued: adopt a stronger Markov condition that guarantees factorization (and hence global independence relations) in all circumstances. Moussouris [16] has identified one such condition, where a system is *strongly Markovian* in case a Markov condition holds for the system and suitable subsystems. That (very!) strong condition forces zero probabilities to be, in a sense, localized, so that probabilities satisfy a nice factorization property. Alas, the condition cannot be guaranteed for all graphs, and its consequences have not been explored in depth so far.

6. Directed Cycles

As noted already, existence of a factorization is a very desirable property for any probabilistic formalism: not only

[4]The Decomposition property states that $X \perp\!\!\!\perp Y \cup W | Z$ implies $X \perp\!\!\!\perp Y | Z$ for sets of random variables W, X, Y, Z [13].

it simplifies calculations, but it also emphasizes modularity in modeling and ease of understanding. In the previous section we have shown that LCNs whose structure is a chain graph do have, under a positivity assumption, a well-known factorization property. We now examine how that result might be extended when structures have directed cycles.

The LMC(LCN) is a local condition that can be applied even in the presence of directed cycles. However, local Markov conditions may not be very satisfactory in the presence of directed cycles, as a simple yet key example suggests:

Example 4 *Take a LCN whose primal graph is a long cycle $A_1 \to A_2 \to \dots A_k \to A_1$, for some large k. No A_i has any non-descendant non-parent. And no A_i has any non-strict-descendant non-parent distinct from A_i. The local Markov conditions we have contemplated do not impose* any *independence relation.*

Local conditions seem too weak when there are long directed cycles. On the other hand, a global condition may work fine in those settings. For instance, apply the GMC(C) to the graph in Example 4; the condition does impose non-trivial independence relations such as $A_1 \perp\!\!\!\perp A_3, \dots, A_{k-1} | A_2, A_k$ and $A_2 \perp\!\!\!\perp A_4, \dots, A_k | A_1, A_3$ (and more generally, for any A_i with $2 < i < k-2$, we have $A_i \perp\!\!\!\perp A_1, \dots, A_{i-2}, A_{i+2}, A_k | A_{i-1}, A_{i+1}$).

At this point it is mandatory to examine results by Spirtes [18], as he has studied local *and* global conditions for directed graphs, obtaining factorization results even in the presence of directed cycles. Spirtes examines the same local Markov condition [18] usually applied to directed acyclic graphs (Section 2):

Definition 11 (LMC(D)) *A node A is independent, given its parents, of all nodes that are not A itself nor descendants nor parents of A.*

Spirtes then notes that this local condition is *not* equivalent to the GMC(C) in Definition 2, and decides to adopt the latter global condition so as to obtain a factorization [18]. He shows [18, Lemma 3] that, for a directed graph that may have directed cycles, a positive probability distribution over the random variables is a product of factors, one per random variable, iff the distribution satisfies the GMC(C) for the graph.

The following example shows that indeed the LMC(D) and the GMC(C) are different in the presence of directed cycles and, more importantly for us, that the GMC(C) for structures is not the same as the GMC(C) that one might adapt to primal graphs so as to obtain Spirtes' results on factorization.

Example 5 *Suppose we have a LCN whose primal graph is depicted in Figure 1.d. For instance, we might have $0.1 \leq \mathbb{P}(X|Y) \leq 0.2$ whenever $Y \to X$ is an edge in that figure. Assume all configurations have positive probability.*

For the primal graph depicted in Figure 1.d, the LMC(LCN) yields only $A \perp\!\!\!\perp C$, $B \perp\!\!\!\perp C | A, D$ and $A \perp\!\!\!\perp D | B, C$. These are the same independence relations produced by the LMC(D) on that graph. However, if we apply the GMC(C) directly to that same graph, we do not *get the same independence relations: then we only obtain $A \perp\!\!\!\perp C$ and $A \perp\!\!\!\perp C | B, D$, perhaps a surprising result (in this case, the graph $\mathcal{G}^{ma}(\{A, B, C, D\})$ is depicted in Figure 2.d).*

Now consider the structure of the LCN; this is the chain graph in Figure 1.e. The LMC(C), and also the LMC(C-STR), yield only $A \perp\!\!\!\perp C$, $B \perp\!\!\!\perp C | A, D$ and $A \perp\!\!\!\perp D | B, C$ when applied to the structure. Moreover, the GMC(C) applied to the structure yields these same independence relations. Clearly this is not the same set of independence relations imposed by the GMC(C) applied to the primal graph. There is a difference between undirected and bi-directed edges when it comes to the GMC(C).

It is worth summarizing the discussion so far. First, it is well-known that the LMC(C) and the GMC(C) are equivalent, under a positivity assumption, in chain graphs. Both conditions may differ in the presence of directed cycles. Second, we know that the LMC(LCN) for primal graphs is equivalent to the LMC(C-STR) with respect to the corresponding structures. And if the structure is a chain graph, then the LMC(C-STR) and the LMC(C) are equivalent when applied to the structure. But for general primal graphs any local condition seems quite weak. We might move to general primal graphs by adapting the GMC(C) to them, so as to look for a factorization result (following Spirtes); however, we saw that the result is not equivalent to what we obtained by applying the GMC(C) to structures (and not equivalent to the original semantics for LCNs when the structure is a chain graph).

In the next section we examine alternative semantics that are based on applying the GMC(C) to structures (possibly with directed cycles). Before we jump into that, it is worth noticing that there are many other relevant results in the literature besides the ones by Spirtes. For instance, *dependency networks* [11] allow for directed cycles and do have a modular specification scheme; they have only an approximate factorization, but that may be enough in applications. Another proposal has been advanced by Schmidt and Murphy [17], where directed cycles are allowed and the adopted Markov condition looks only at the *Markov blanket* of nodes; it does not seem that a factorization has been proven for that proposal, but it is attractive in its simplicity. There are also many kinds of graphs that have been contemplated to handle causal loops and dynamic feedback systems [3, 4, 12]. This is indeed a huge literature, filled with independence conditions and factorization properties, to which we cannot do justice in the available space. It is necessary to examine whether we can bring elements of those previous efforts into LCNs. We leave a more detailed study for the future.

7. New Semantics for LCNs

In this section we explore new semantics for LCNs by applying the GMC(C) to structures. This is motivated by the weakness of local conditions as discussed in the previous section, and also by the fact that a condition based on moralized graphs is the most obvious route to factorization properties (as the Hammersley-Clifford theorem can then be invoked under a positivity assumption [16]).

Here is a (new) semantics: a LCN represents the set of probability distributions over its nodes such that all constraints in the LCN are satisfied, and each distribution satisfies the GMC(C) with respect to the structure. Note that the GMC(C) is equivalent to the LMC(LCN) when a structure is a chain graph, but these conditions may differ in the presence of directed cycles (Example 4).

The path to a factorization result is then as follows. Take the structure and, for each node A, build a set C_A with all nodes that belong to directed cycles starting at A. If there is a directed cycle in a set C_B such that B is in C_A, then merge C_A and C_B into a set $C_{A,B}$; repeat this until there are no more sets to merge (this must stop, in the worst case with a single set containing all nodes). For each set, replace all nodes in the set by a single "super"-node, directing all edges in and out of nodes in the set to this super-node. The resulting graph has no directed cycles, so the GMC(C) applied to it results in the usual factorization over chain components of the resulting graph. Now each super-node is in fact a set of nodes that can be subject to further factorization, even though it is an open question whether a decomposition can be obtained with factors that are directly related to graph properties.

To continue, we suggest that, instead of using structures as mere secondary objects that help us clarify the meaning of primal graphs, structures should be the primary tools in dealing with LCNs. That is, we should translate every LCN to its structure (without going through the primal graph) and then apply appropriate Markov conditions there. Given a LCN, with the same sets of constraints as specified previously, we can build its structure by taking every proposition as a node and then:

1. For each constraint $\alpha \leq \mathbb{P}(\phi|\varphi) \leq \beta$ in $\mathcal{T}_U$, add an undirected arrow between each pair of proposition-nodes in ϕ.

2. For each constraint $\alpha \leq \mathbb{P}(\phi|\varphi) \leq \beta$ add a directed edge from each proposition-node in φ to each proposition-node in ϕ (if φ is $\top$, there is no such edge to add).

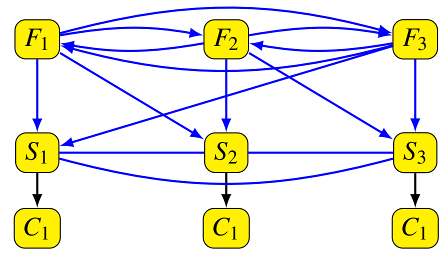


Figure 6: The mixed-structure for the LCN in Example 1.

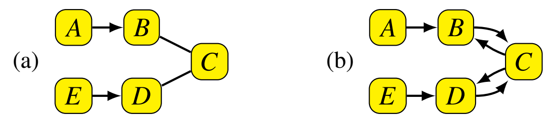


Figure 7: Structures and mixed-structures in Example 6.

3. Remove multiple identical edges.

4. For each pair of nodes A and B, if there is a bi-directed edge $A \leftrightarrows B$ between them, replace the two edges by a single undirected edge $A \sim B$.

For instance, the procedure above goes directly from the LCN in Example 1 to the structure in Figure 4.b.

When we think of structures this way, we might wish to differentiate the symmetric connections that surface when a pair of propositions appear in a formula ϕ from mutual influences when one proposition is conditioned on the other and vice-versa. An alternative semantics would then be as follows. Take a LCN and build a *mixed-structure* by going through the first three steps above. That is, create a node per proposition that appears in the LCN; then take each constraint in $\mathcal{T}_U$ and add undirected edges between any two propositions in ϕ, and finally take each constraint and add a directed edge from each proposition that appears in φ to each proposition that appears in the corresponding ϕ. Figure 6 depicts the mixed-structure for Example 1.

Now adopt: a LCN represents the set of probability distributions over its nodes such that all constraints in the LCN are satisfied, and each distribution satisfies the GMC(C) with respect to the mixed-structure.

The next example emphasizes the differences between semantics.

Example 6 *Suppose we have a LCN with constraints* $\mathbb{P}(B|A) = 0.2$, $\mathbb{P}(D|E) = 0.3$, $\mathbb{P}(B \vee C) = 0.4$, $\mathbb{P}(C \vee D) = 0.5$. *Both the structure and the mixed-structure of this LCN is depicted in Figure 7.a. Consider another LCN with constraints* $\mathbb{P}(B|A \wedge C) = 0.2$, $\mathbb{P}(C|B \wedge D) = 0.3$, *and* $\mathbb{P}(D|C \wedge E) = 0.4$. *This second LCN has the same structure as the first one, but the mixed-structure are depicted in Figure 7.b. The GMC(C) produces quite different* *sets of independence relations when applied to these distinct mixed-structures; for instance,* $A,B \perp\!\!\!\perp D|C,E$ *in the first LCN, but not necessarily in the second;* $A,B \perp\!\!\!\perp E|C,D$ *in the second LCN, but not necessarily in the first. This seems appropriate as the LCNs convey quite distinct scenarios, one related to the symmetry of logical constraints, the other related to the links induced by directed influences.*

We hope to pursue a comparison between the theoretical and pragmatic aspects of these semantics in future work.

8. Conclusion

In this paper we visited many Markov conditions that can be applied, if properly adapted, to Logical Credal Networks [14]. We reviewed existing concepts and introduced the notion of structure of a LCN, showing that the original local condition LMC(LCN) can be viewed as a local condition on structures. We then showed that the LMC(LCN) is equivalent to a usual local condition when the structure is a chain graph, and this leads to a factorization result. Moreover, we introduced a new semantics based on structures and a global Markov condition — a semantics that agrees with the original one when the structure is a chain graph but that offers a possible path to factorization properties.

There are many issues left for future work. LCNs stress the connection between the syntactic form of constraints and the semantic consequences of independence assumptions, a theme that surfaces in many probabilistic logics. We must investigate more carefully the alternatives when extracting independence relations from constraints, in particular to differentiate ways in which bi-directed edges are created.

We must also examine positivity assumptions. What is the best way to guarantee a factorization? Should we require the user to explicitly express positivity assumptions? Should we allow for logical constraints that assign probability zero to some configurations; if so, which kinds of configurations? How to make such constraint-induced zero probabilities compatible with factorization properties?

It is also important to study a large number of Markov conditions that can be found in the literature but that we have skipped here, both conditions connected with chain graphs and conditions connected with causal and feedback models. We must verify which conditions lead to factorization results, and which conditions are best suited to capture the content of logical formulas, causal influences, feedback loops.

In a more applied perspective, we must investigate whether the ideas behind LCNs can be used with practical specification languages such as Probabilistic Answer Set Programming, and we must test how various semantics for LCNs fare in realistic settings.

Acknowledgements

This work was carried out at the Center for Artificial Intelligence (C4AI - USP/IBM/FAPESP), with support by the São Paulo Research Foundation (FAPESP grant 2019/07665-4) and by the IBM Corporation. The author was partially supported by CNPq grants 312180/2018-7 and 305753/2022-3. We acknowledge support by CAPES - Finance Code 001.

References

[1] Kim A. Andersen and John N. Hooker. A linear programming framework for logics of uncertainty. *Decision Support Systems*, 16:39–53, 1996.

[2] Fahiem Bacchus. *Representing and Reasoning with Probabilistic Knowledge: A Logical Approach*. MIT Press, Cambridge, 1990.

[3] Christel Baier, Clemens Dubslaff, Holger Hermanns, and Nikolai Kafer. On the foundations of cycles in Bayesian networks. In J. F. Raskin, K. Chatterjee, L. Doyen, and R. Majumdar, editors, *Principles of Systems Design*, volume 13660 of *Lecture Notes in Computer Science*. Springer, 2022.

[4] Stephan Bongers, Patrick Forré, Jonas Peters, and Joris M. Mooij. Foundations of structural causal models with cycles and latent variables. *Annals of Statistics*, 49(5):2885–2915, 2021.

[5] Robert G. Cowell, A. Philip Dawid, Steffen L. Lauritzen, and David J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag, New York, 1999.

[6] Fabio G. Cozman. Credal networks. *Artificial Intelligence*, 120:199–233, 2000.

[7] Fabio Gagliardi Cozman and Rodrigo Bellizia Polastro. Complexity analysis and variational inference for interpretation-based probabilistic description logics. In *Conference on Uncertainty in Artificial Intelligence*, pages 117–125, Corvallis, Oregon, 2009. AUAI Press.

[8] José Carlos Ferreira da Rocha and Fabio Gagliardi Cozman. Inference in credal networks: branch-and-bound methods and the A/R+ algorithm. *International Journal of Approximate Reasoning*, 39(2-3):279–296, 2005.

[9] D. Doder and Z. Ognjanovic. Probabilistic logics with independence and confirmation. *Studia Logica*, 105: 943–969, 2017.

[10] Joseph Y. Halpern. *Reasoning about Uncertainty*. MIT Press, Cambridge, Massachusetts, 2003.

[11] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.

[12] A. Hyttinen, F. Eberhardt, and P. Hoyer. Learning linear cyclic causal models with latent variables. *Journal of Machine Learning Research*, 13:3387–3439, 2012.

[13] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[14] Radu Marinescu, Haifeng Qian, Alexander Gray, Debarun Bhattacharjya, Francisco Barahona, Tian Gao, Ryan Riegel, and Pravinda Sahu. Logical credal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[15] Denis Deratani Mauá and Fabio Gagliardi Cozman. Thirty years of credal networks: Specification, algorithms and complexity. *International Journal of Approximate Reasoning*, 126:133–157, 2020.

[16] John Moussouris. Gibbs and Markov random systems with constraints. *Journal of Statistical Physics*, 10 (1):11–33, January 1974. ISSN 0022-4715 (Print) 1572-9613 (Online).

[17] Mark Schmidt and Kevin Murphy. Modeling discrete interventional data using directed cyclic graphical models. In *Conference on Uncertainty in Artificial Intelligence*, 2009.

[18] Peter Spirtes. Directed cyclic graphical representations of feedback models. In *Conference on Uncertainty in Artficial Intelligence*, pages 491–498, 1995.