Chapter 10
Chain Graphs and Gene Networks

Dag Sonntag and Jose M. Peña

Abstract Chain graphs are graphs with possibly directed and undirected edges, and no semidirected cycle. They have been extensively studied as a formalism to represent probabilistic independence models, because they can model symmetric and asymmetric relationships between random variables. This allows chain graphs to represent a wider range of systems than Bayesian networks. This in turn allows for a more correct representation of systems that may contain both causal and non-causal relationships between its variables, like for example biological systems. In this chapter we give an overview of how to use chain graphs and what research exists on them today. We also give examples on how chain graphs can be used to model advanced systems, that are not well understood, such as gene networks.

10.1 Introduction

In the previous chapter we saw how we could model advanced systems as Bayesian networks (BNs) by representing the causal relations between the variables in the system as directed edges. These models are widely used today but as noted in the previous chapter they do have certain shortcomings. In this chapter we will discuss one such shortcoming, namely the inability to model non-causal relations, and how this can be solved using more expressive probabilistic graphical model (PGM) classes such as chain graphs (CGs).

When an expert is modelling a system it is often relatively easy to find causal relations between the variables in the system and thereby model it as a BN. This is especially true for well known systems where all relevant factors are included as variables in the model. However, for more advanced systems some relations between directly correlated variables might not have such a clear causal structure. This can be for many reasons, such as that a hidden common cause exists between the variables or that there exist selection bias between them. Modelling these relations with directed edges is then incorrect from the perspective of interpretation and can cause incorrect reasoning subsequently.

A. Hommersom and P.J.F. Lucas (eds.), *Biomedical Knowledge Representation*, LNAI 9521, DOI 10.1007/978-3-319-28007-3_10

CGs solve this problem by extending the ideas of BNs with an additional type of edge representing non-causal relations between variables. Representing variables as nodes, causal relations with directed edges and non-causal relations with non-directed edges these models can therefore represent a larger set of models than BNs. At the same time CGs keep key features of BNs such as their interpretability and efficiency when it comes to inference and structure learning.

CGs are also interesting because they correctly can represent a much larger set of independence models, and thereby probability distributions, than BNs, Markov networks (MNs) or covariance graphs (covGs). BNs, MNs and covGs are the PGM classes most commonly used today when modelling bioinformatics systems. This means that for a probability distribution p there may be no BN G able to represent only and all independences in p when a CG F can. A BN can represent any probability distribution, but only by including fewer independences, and thereby additional dependences, than what actually exist in the underlying probability distribution. These spurious, additional, dependences can then later be "removed" by the correct parametrization, but this is still problematic for several reasons. Firstly, the advantage of using PGMs, such as the speed of inference, is larger the sparser the graph is. By having more edges than necessary this advantage is lost. Secondly, some of these edges might not make sense from a biological point of view. This is problematic for practitioners trying to understand the system through its graph, since the edges obscure the true (in)dependences between the variables.

A problem with CGs is however that there exists multiple types of non-causal relations as described above. This means that depending on what kind of non-causal relation we mean with the non-directed edge in our models we represent different systems and thereby independence models. To distinguish the different meanings of the non-directed edge we say that we have different CG interpretations, and that the non-directed edge is interpreted differently in different CG interpretations. Today there exists mainly three CG interpretations in research. These are the Lauritzen-Wermuth-Frydenberg (LWF) interpretation [7, 13], the Andersson-Madigan-Perlman (AMP) interpretation [1] and the multivariate regression (MVR) interpretation [3, 4].

One question that can be asked is how much more expressive CGs are compared to BNs? If the advantage is small the additional complexity might not translate into significantly better models. It has however been shown that as the number of variables increases CGs can express exponentially many more independence models compared to BNs. So for only 20 variables any CG interpretation can express approximate 1000 times more independence models, and thereby systems, compared to BNs [25, 26]. Hence for large domains with hundreds of variables the number of independence models representable by BNs is incredibly small compared the number of independence models representable by CGs. Therefore, CGs are much more likely to provide a realistic graph structure instead of obscuring the true relations in the system [25, 26].

In the rest of this chapter we will cover how these different CG interpretations work and what systems they can represent. First, in the next section, we will however describe the notation we use. In Sect. 10.3 we then describe the background and meaning of the different CG interpretations, while in Sect. 10.4 we describe how such a CG

can be learnt from a probability distribution. After a short conclusion and summary in Sect. 10.5, we provide an alternative illustration of CGs as systems of linear equations in the Appendix. For simplicity we limit our discussion to continuous variables but most results can also be generalised to systems with discrete or mixed variables.

10.2 Background and Notation

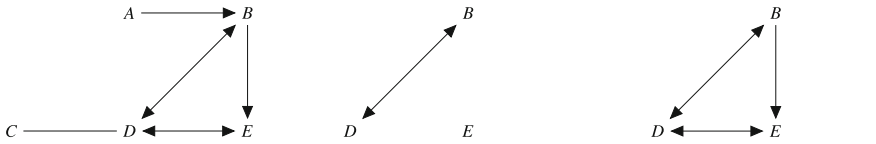
In this section, we review some concepts from PGMs that are used later in this chapter. All graphs and probability distributions are defined over a finite set of variables V represented as nodes in the graphs.

If a graph G contains an edge between two nodes V_1 and V_2, we denote with $V_1 \rightarrow V_2$ a *directed edge*, with $V_1 \leftrightarrow V_2$ a *bidirected edge* (sometimes also called a *dashed edge*) , and with $V_1 - V_2$ an *undirected edge*. With a *non-directed* edge we mean either a bidirected edge or undirected edge. A set of nodes is said to be *complete* if there exists edges between all pairs of nodes in the set. A complete set of nodes is said to be a *clique* if there exists no superset of it that is complete.

The *parents* of a set of nodes X of G is the set $pa_G(X) = \{V_1 | V_1 \rightarrow V_2$ is in G, $V_1 \notin X$ and $V_2 \in X\}$. The *children* of X is the set $ch_G(X) = \{V_1 | V_2 \rightarrow V_1$ is in G, $V_1 \notin X$ and $V_2 \in X\}$. The *spouses* of X is the set $sp_G(X) = \{V_1 | V_1 \leftrightarrow V_2$ is in G, $V_1 \notin X$ and $V_2 \in X\}$. The *neighbours* of X is the set $nb_G(X) = \{V_1 | V_1 - V_2$ is in G, $V_1 \notin X$ and $V_2 \in X\}$. The *boundary* of X is the set $bd_G(X) = pa_G(X) \cup nb_G(X) \cup sp_G(X)$. The *adjacents* of X is the set $ad_G(X) = \{V_1 | V_1 \rightarrow V_2, V_1 \leftarrow V_2$, $V_1 \leftrightarrow V_2$ or $V_1 - V_2$ is in G, $V_1 \notin X$ and $V_2 \in X\}$.

To exemplify these concepts we can study the graph G with five nodes shown in Fig. 10.1a. In the graph we can see two bidirected edges, one between B and D and one between D and E. Hence we know the spouses of D are B and E. G also contains two directed edges from A to B and from B to E and we can see that E is the only child of B and B is the only child of A. Finally G also contains one undirected edge between C and D and hence C is a neighbour of D. All and all this means that the boundary of B is A and D while the adjacents of B also contains E in addition to A and D.

A *route* from a node V_1 to a node V_n in G is a sequence of nodes $V_1, \ldots, V_n$ such that $V_i \in ad_G(V_{i+1})$ for all $1 \leq i < n$. A *path* is a route containing only distinct nodes. The length of a path is the number of edges in the path. A path is called a *cycle* if $V_n = V_1$. A path is *descending* if $V_i \in pa_G(V_{i+1}) \cup sp_G(V_{i+1}) \cup nb_G(V_{i+1})$ for all $1 \leq i < n$. The *descendants* of a set of nodes X of G is the set $de_G(X) = \{V_n |$ there is a descending path from V_1 to V_n in G, $V_1 \in X$ and $V_n \notin X\}$. A path is *strictly descending* if $V_i \in pa_G(V_{i+1})$ for all $1 \leq i < n$. The *strict descendants* of a set of nodes X of G is the set $sde_G(X) = \{V_n |$ there is a strictly descending path from V_1 to V_n in G, $V_1 \in X$ and $V_n \notin X\}$. The *ancestors* (resp. *strict ancestors*) of X is the set $an_G(X) = \{V_1 | V_n \in de_G(V_1), V_1 \notin X, V_n \in X\}$ (resp. $san_G(X) = \{V_1 | V_n \in sde_G(V_1), V_1 \notin X, V_n \in X\}$). Note that the definition for strict descendants given here coincides to the definition of descendants given by Richardson [21].



(a) A graph G (b) A subgraph of G over $\{B,D,E\}$ (c) The subgraph of G induced by $\{B,D,E\}$

Fig. 10.1 Three different graphs

A cycle is called a *semi-directed cycle* if it is descending and $V_i \to V_{i+1}$ is in G for some $1 \leq i < n$.

To exemplify these concepts we can once again look at the graph G in Fig. 10.1a. We can here see two paths between B and C, $B \leftrightarrow D - C$ and $B \to E \leftrightarrow D - C$, and that the latter of these is descending while the former is not. An example of a route between B and C that is not a path is $B \leftrightarrow D \leftrightarrow E \leftarrow B \leftrightarrow D - C$. We can see that G contains one cycle $B \leftrightarrow D \leftrightarrow E \leftarrow B$ that is semi-directed. Moreover we can see that E is a strict descendant of A due to the strictly descending path $A \to B \to E$, while D is not. D is however in the descendants of A together with B, C and E. A is therefore an ancestor of all variables except itself.

A Markov network (MN) (resp. covariance graph (covG)) contains only undirected (resp. bidirected) edges while a BN only contains directed edges and no semi-directed cycles. A CG under the Lauritzen-Wermuth-Frydenberg (LWF) interpretation, denoted *LWF CG*, contains only directed and undirected edges but no semi-directed cycles. Likewise a CG under the Andersson-Madigan-Perlman (AMP) interpretation, denoted *AMP CG*, is a graph containing only directed and undirected edges but no semi-directed cycles. A CG under the multivariate regression (MVR) interpretation, denoted *MVR CG*, is a graph containing only directed and bidirected edges but no semi-directed cycles. A *chain component* C of a LWF CG or an AMP CG (resp. MVR CG) is a maximal set of nodes such that there exists a path between every pair of nodes in C containing only undirected edges (resp. bidirected edges). A *subgraph* of G is a subset of nodes and edges in G. A subgraph of G induced by a set of its nodes X is the graph over X that has all and only the edges in G whose both ends are in X.

If we go back to our example in Fig. 10.1 we can see that the graph in Fig. 10.1b is a subgraph of G over the variables B, D and E while the graph in Fig. 10.1c is a subgraph induced by the same variables. We can also see that G is not a CG of any of the interpretations since it contains a semi-directed cycle. An example of a LWF CG or an AMP CG is instead shown in Fig. 10.2a while an example of a MVR CG is shown in Fig. 10.2b. We can here see that H contains three connectivity components $\{A\}$, $\{B\}$ and $\{C, D\}$ and that F contains two connectivity components $\{A\}$ and $\{B, C, D\}$.

Let X, Y and Z denote three disjoint subsets of V. We say that X is *conditionally independent* from Y given Z if the value of X does not influence the value of Y when

Fig. 10.2 Two different CGs

(a) A LWF or AMP CG H (b) A MVR CG F

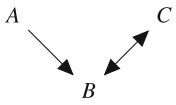
the values of the variables in Z are known, i.e. $p(X, Y|Z) = p(X|Z)p(Y|Z)$ holds and $p(Z) > 0$. We denote this by $X \perp_p Y|Z$ if it holds in a probability distribution p while we with $X \not\perp_p Y|Z$ mean that it does not hold in p. Moreover we say that X is separated from Y given Z in a graph G if the separation criterion of G represents that X is conditionally independent of Y given Z. We denote the this by $X \perp_G Y|Z$ and we will discuss different separation criteria for CGs later in this chapter. Similarly we denote with $X \not\perp_G Y|Z$ that the separation criterion of G does not represent the conditional independence. A probability distribution p is said to fulfill the *global Markov property* with respect to a graph G, if for any $X \perp_G Y|Z$, given the separation criterion for the PGM class to which G belongs, $X \perp_p Y|Z$ holds. The *independence model* M induced by a probability distribution p (resp. a graph G), denoted as $I(p)$ (resp. $I(G)$), is the set of statements $X \perp_p Y|Z$ (resp. $X \perp_G Y|Z$) that holds in p (resp. G). Given two independence models M and N, we say that N includes M ($M \subseteq N$), iff $X \perp_M Y|Z$ implies that $X \perp_N Y|Z$ for every X, Y and Z.

We say that a probability distribution p is *faithful* to a graph G when $X \perp_p Y|Z$ iff $X \perp_G Y|Z$ for all X, Y and Z. We say that two graphs G and H are *Markov equivalent* or that they are in the same *Markov equivalence class* iff $I(G) = I(H)$. A graph G is *inclusion optimal* for a probability distribution p if $I(G) \subseteq I(p)$ and if there exists no other graph H in the PGM class of G such that $I(G) \subset I(H) \subseteq I(p)$.

To illustrate the last concepts we can look at the MVR CG J and the independence models in Fig. 10.3. In Fig. 10.3b we can see the independences that hold in J and hence the independence model of J. Finally we can also see another independence model in Fig. 10.3c such that $I(J) \subseteq M$ and hence that M includes the independence model represented by J.

10.3 CG Interpretations

The research on CGs started in the late 1980s with the Lauritzen-Wermuth-Frydenberg (LWF) interpretation in order to combine BNs and MNs into more expressive models. Subsequently, the Andersson-Madigan-Perlman (AMP) interpretation and the multivariate regression (MVR) interpretation, both in common use in recent literature, were proposed. Each interpretation is based on a different separation criterion and a different interpretation of the edges. No interpretation subsumes another [5, 23], and no interpretation is generally better than any other. LWF, AMP and MVR interpretations are



(a) A MVR CG J

$A \perp_J C | \emptyset$
$C \perp_J A | \emptyset$

(b) The independence model of J

$A \perp C | \emptyset$
$C \perp A | \emptyset$
$A \perp C | B$
$C \perp A | B$

(c) Another independence model M

Fig. 10.3 Example of independence models

just different from each other, similarly as BNs and MNs are different from each other, and are suited to different problems. We will in this chapter present each interpretation in three different ways. First in the classical sense, i.e. in terms of their separation criteria as in Drton [5], secondly in terms of systems of linear equations and third with some intuitive meaning behind the edges in the CGs. Finally we will also give examples of how they can be used. Moreover, in the next section we will discuss how to decide which interpretation to use when modelling a system with CGs.

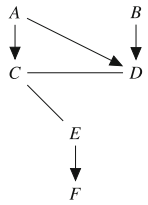
First we will however see how BNs are presented in these three ways. For BNs the separation criterion is as follows. Given three disjoint sets of nodes X, Y and Z in a BN G, $X \perp_G Y | Z$ iff there exists no path between X and Y such that:

1. every non-collider on the path is not in Z and
2. every collider on the path is in Z or $san_G(Z)$.

A node B is said to be a *collider* between two nodes A and C on a path if the following configuration exists in the path: $A \rightarrow B \leftarrow C$. For any other configuration the node B is a non-collider on the path. In addition, the interpretation in terms of a system of linear equations is as follows. The probability distribution of every node in a BN depends only on its parents. This means that every node X_i is modelled by the equation $X_i = \beta_i * pa_G(X_i) + \epsilon^i$ in the associated system of linear equations, where β_i is a weight vector measuring the influence of the individual parents and the noise $\epsilon^i \sim \mathcal{N}(0, \sigma_i)$ is independent of any other node's noise. The intuitive meaning is simply that the parent nodes are the cause of the children nodes.

For CGs the different interpretations have different separation criteria. As noted in the introduction, the feature all CGs share is that they contain subgraphs, called chain components, that are connected to each other by directed edges. Within each chain component the type of edges varies depending on the interpretation: LWF CGs and AMP CGs contain undirected edges while MVR CGs contain bidirected edges. Even though the intuitive meaning of a CG is not as simple as for a BN, there are similarities between the two PGM classes. For example, the separation statements encoded by a CG correspond to the non-existence of routes with certain features, as in BNs. Moreover, in terms of linear equations each component of a CG can be seen as a supernode, with the corresponding probability distribution determined only by its parents. If we let K_i be the component i in a CG G, then G has an associated system of linear equations with normally distributed errors as follows:

$$K_i = \beta_i \; pa_G(K_i) + \epsilon^i \qquad \text{where} \qquad \epsilon^i \sim \mathcal{N}(0, \Lambda^i).$$



(a) An example CG G

$A \perp_G B|\emptyset$
$B \not\perp_G C|\emptyset$
$A \not\perp_G B|C$
$A \perp_G F|E$
$B \not\perp_G E|D$

(b) LWF CG separations in G

$A \perp_G B|\emptyset$
$B \perp_G C|\emptyset$
$A \perp_G B|C$
$A \perp_G F|E$
$B \not\perp_G E|D$

(c) AMP CG separations in G

Fig. 10.4 An example CG G and some corresponding separations according to the LWF and AMP interpretations.

ϵ^i represents the noise, or influence, between the nodes in the same component. How this noise and the β_i-vector are modelled varies between the different interpretations, and gives them different intuitive meanings.

10.3.1 The LWF Interpretation

The LWF interpretation was introduced by Lauritzen, Wermuth and Frydenberg in 1989 [7, 13] and is the most well researched CG interpretation. As noted above, LWF CGs contain components that are connected to each other by directed edges. The separation criterion is the following. Given three disjoint subsets of nodes X, Y and Z in a LWF CG G, $X \perp_G Y|Z$ iff there exists no route between X and Y such that:

1. every node in a non-collider section on the route is not in Z and
2. some node in every collider section on the route is in Z.

A *section* of a route is a maximal non-empty set of nodes $B_1 ... B_n$ such that the route contains the subroute $B_1 - B_2 - \ldots - B_n$. It is called a *collider section* if $B_1 \ldots B_n$ together with the two neighbouring nodes in the route, A and C (note that A and C might be the same node), form the subroute $A \rightarrow B_1 - B_2 - \ldots - B_n \leftarrow C$ in the route. For any other configuration the section is a non-collider section.

A simple example of a CG is shown in Fig. 10.4a. Here the CG has four chain components: A, B, $\{C, D, E\}$ and F. If the graph is interpreted as a LWF CG the separations and non-separations shown in Fig. 10.4b hold. Note that these are not all the separations that hold in G.

When reasoning in terms of linear equations, the parents of a component can be interpreted as the causes of the nodes in that component, and directed edges have the same meaning as in a BN. So the linear equation of a node X_j in a LWF CG is $X_j = \beta_j \; pa_G(K_i) + \epsilon^j$ where K_i is the component to which X_j. As shown in the Appendix, the k-th element of β_j can be interpreted as the sum of the weights of all the paths in G between the parent X_k of K_i and the node X_j of K_i such that the

nodes in these paths are all in $X_k \cup K_i$, and where the path weight itself is the product of the weight of its edges. The noise ϵ^j is then determined by the associated inverse covariance matrix of that component. Furthermore, the corresponding entry in the inverse covariance matrix for two nodes X_j and X_m can be non-zero iff there exists an undirected edge $X_j - X_m$ in G (see the Appendix for details). For example, we can see from Fig. 10.4a that the influence from node B onto node D is direct since there only exists one path between them. However, the influence from node A onto node E is determined by the path $A \to C - E$ as well as $A \to D - C - E$ (see the Appendix for details).

This characterisation of the influence of a parent of K_i means that parents influence all the nodes in K_i, as influence propagates to all of K_i through its undirected edges. We can see, for example, that in the second example above the influence from A onto E is the same as A onto C except for the last path between C and E. This makes LWF CGs similar to module networks, another PGM class that has shown promising results for gene networks [22]. In module networks every node in a module, which is similar to a component, has the same parents and parameters. In a LWF CG, every node in the same component have the same parents when the LWF CG is seen as a system of linear equations. However, the influence of the parents on a node depends on the paths between them and, thus, it may be different for different nodes in the component.

An example of a situation when LWF CGs are useful is when we want to model a system with knowledge obtained from several experts, each with his or her own exclusive field of competence. Each expert then gives information about the structural relationships between the variables within his or her domain given outside factors that affect the variables in his or her domain of expertise. The expert does this by providing a MN over the variables in the domain and their outside factors. Moreover, since the expert only knows about his or her domain and not how the outside factors are related, he or she must assume that all outside factors are adjacent when creating the MN. The subgraph of the MN induced by the variables in the experts domain can then be seen as a component in a resulting LWF CG while the outside factors are added as parents to their previous neighbours in the component. The internal structure of the outside factors will be defined by some other expert, who is expert over that domain. If a strict causal ordering is kept between the variables, putting the different chain components together into a single graph then results in LWF CG [28]. An example of this in medicine can be if we have three experts, one expert modelling the probability that a person have certain gene-expressions in his or her DNA, one that models the probability of different protein signalling data occurring in blood samples given these gene-expressions and one that models the occurrence of different traits, such as diseases, given the gene-expressions.

Other settings in which LWF CGs are appropriate is for modelling the equilibrium state of a system containing feedback loops [12] or when variables of a system only can be measured in an aggregated state [6]. It can also be noted that if a LWF CG only contains directed edges it can be read as a BN while if it only contains undirected edges it can be read as a MN.

10.3.2 The AMP Interpretation

The AMP CG interpretation was introduced by Andersson, Madigan and Perlman [1] as an alternative to the LWF interpretation because it preserves the recursive characteristics of BNs. Similarly to LWF CGs, AMP CGs also contain components connected to each other by directed edges, whereas each component internally only contains undirected edges. As a result, an AMP CG containing only directed edges can be read as a BN and an AMP CG containing only undirected edges can be read as a MN similarly as a LWF CG. However, the separation criterion is different compared to LWF CGs. Given three disjoint subsets of nodes X, Y and Z in an AMP CG G, $X \perp_G Y|Z$ iff there exists no route between X and Y such that:

1. every non-collider on the route is not in Z and
2. every collider on the route is in Z or $san_G(Z)$.

A node B is said to be a *collider* in an AMP CG G between two nodes A and C on a route if one of the following configurations exists in G: $A \rightarrow B \leftarrow C$, $A \rightarrow B - C$ or $A - B \leftarrow C$. For any other configuration the node B is a non-collider. In the case of the CG shown in Fig. 10.4a, we can see that the separations and non-separations in Fig. 10.4c hold if we interpret it as an AMP CG. Note that these are not all the separations and non-separations that hold in G.

The modelling of the noise also differs from LWF CGs. In the Appendix it is shown that the associated linear equation of a node X_j in an AMP CG G is $X_j = \beta_j \, pa_G(X_j) + \epsilon^j$. The node depends only on its parents and not on the parents of the whole component, as it does in the case of LWF CGs. The noise ϵ^j is then controlled by the inverse covariance matrix of that component. Furthermore, the corresponding entry in the inverse covariance matrix for two nodes X_j and X_k can be non-zero iff there exists an undirected edge $X_j - X_k$ in G (see the Appendix for details). Intuitively, a small set of nodes works as an interface between other nodes in the component and its parents. For example, we can see that C and D in Fig. 10.4a block the influence from the parents A and B onto E if the graph is interpreted as an AMP CG.

AMP CGs are useful when we have a set of variables for which the internal relations has no causal ordering, so the relations should be modelled as a MN, but also a second set of variables which can be seen as causes for some of these variables in the first set. The internal structure of the first set of variables can then be modelled as a MN, creating a chain component in an AMP CG, and the causes as parents of some of the variables in the chain component. Note that for AMP CGs the parents only affects the direct children in the chain component, not all the nodes in the chain component such as in the case of LWF CGs. An example in medicine when such a model might be appropriate is when we are modelling pain levels on different areas on the body of a patient. The pain levels can then be seen as correlated "geographically" over the body, and hence be modelled as a MN. Certain other factors do however exist that alters the pain levels locally at some of these areas, such as the type of body part the area is located on or if local anaesthetic has been administered in that area and so on. These outside factors can then be modelled as parents affecting the pain levels locally.

While both LWF CGs and AMP CGs consist of MNs as chain components they differ in the way the parents of the component affect the variables in the component. In a LWF CG each parent affects all the variables in the component, i.e. the information travels through the children, while in an AMP CG the parents only affects the actual children, i.e. the information does not travel to the other variables in the chain component. Hence when we have a system for which some parts best are modelled as MNs and some parts as BNs we can use either a LWF CG or AMP CG, depending on which type fits the independence model of the system best.

10.3.3 The MVR Interpretation

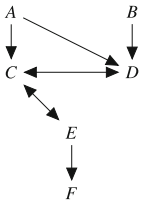
MVR CGs were originally introduced by Cox and Wermuth [3, 4], and are equivalent to the acyclic directed mixed graphs without semi-directed cycles presented by Richardson [21]. Cox and Wermuth represented these graphs using directed edges and dashed edges, but we follow Richardson [21] as we feel that the notation is closer to that of BNs when it comes to the separation criterion.

The most important difference between the MVR CGs compared to AMP CGs and LWF CGs is that MVR CG components contains bidirected instead of undirected edges. As a result, MVR CGs is a superclass of BNs and covGs instead of BNs and MNs as in the case of AMP and LWF CGs [4]. MVR CGs also have the following separation criterion: Given three disjoint subsets of nodes X, Y and Z in a MVR CG G, $X \perp_G Y | Z$ iff there exists no path between X and Y such that:

1. every non-collider on the path is not in Z and
2. every collider on the path is in Z or $san_G(Z)$.

A node B is said to be a *collider* in a MVR CG G between two nodes A and C on a path iff one of the following configurations exists in the path: $A \rightarrow B \leftarrow C$, $A \rightarrow B \leftrightarrow C$, $A \leftrightarrow B \leftarrow C$ or $A \leftrightarrow B \leftrightarrow C$. For any other configuration the node B is said to be a non-collider. An example of a MVR CG is shown in Fig. 10.5c, with some of the corresponding separations and non-separations in Fig. 10.5b.

The associated system of linear equations is similar to that of the AMP CGs: each node depends only on its parents and not on the parents of the whole component.



(a) Example CG G

$A \perp_G B | \emptyset$
$B \perp_G C | \emptyset$
$A \perp_G B | C$
$A \perp_G F | E$
$A \not\perp_G F | C$

(b) MVR CG separations in G

Fig. 10.5 A MVR CG and some corresponding separations.

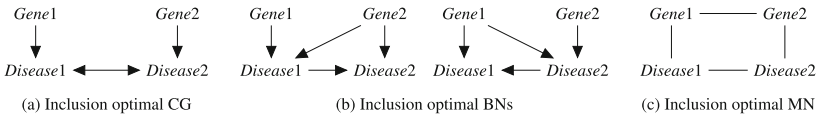


Fig. 10.6 A gene and disease example with MVR CG representation, BN representation and MN representation

The associated linear equation for a node X_j can therefore be written as $X_j = \beta_j\ pa(X_j) + \epsilon^j$, where ϵ^j is dependent on the other nodes in the same component. Unlike AMP CGs, MVR CGs can contain non-zero values in the corresponding covariance matrix (not the inverse covariance matrix as for AMP CGs) only for nodes that are spouses (see the Appendix for details). The intuitive meaning behind the MVR CGs is therefore very close to that of AMP CGs, differing only in the noise modelling.

A typical situation that gives rise to a MVR CG is in the presence of hidden variables, i.e. unobserved variables that are parents of at least two observed variables in the data. An example of a situation for which a MVR CG is useful is if we have a system containing two genes and two diseases caused by these such that *Gene1* is the cause of *Disease1* and *Gene2* is the cause of *Disease2* but where we also can see that the diseases are correlated. In this case we might suspect the presence of an unknown factor inducing the correlation between *Disease1* and *Disease2*, such as being exposed to a stressful environment. Having such a hidden variable results in the independence model described in the information above. We can now choose whether we would like to model this hidden variable in our model, but due to difficulties of measurement let us assume we do not. The MVR CG representing the information above is shown in Fig. 10.6a while the inclusion optimal BNs and MN are shown in Fig. 10.6b and 10.6c, respectively. We can now see that it is only the MVR CG that describes the relations in the system correctly.

10.4 CG Learning

As is the case with BNs, the graph structure of a CG can be learnt either from expert knowledge on the system or from data. The process of creating a CG from expert knowledge is very similar to that of a BN but where the non-directed edges can be used to model the variable correlations described in the previous section. An example of this process is given in Subsect. 10.4.1. In Subsects. 10.4.2 and 10.4.3 we then cover the structure learning algorithms that exist today that allow a CG to be learnt from a probability distribution p. First in the special case where we assume p is faithful to some CG and then the more general case where we do not. Finally, in Subsect. 10.4.4, we also discuss the current research on how CGs can be factorized and how the parameters can be learnt.

10.4.1 Learning CGs by Expert Knowledge

The process of creating a CG from expert knowledge of a domain is very similar to that of creating a BN from expert knowledge. Some important parts do however differ, such as choosing which CG interpretation to use. In this subsection we will therefore give an example of how this process can be performed.

The example we will be using was introduced by Lappenschaar et al. [10] and concerns the interaction between two diseases, *diabetes mellitus* and *lipid disorder*, along with typical blood measurements, two risk factors and a possible treatment. The blood measurements we are considering are *elevated blood cholesterol levels* and *elevated blood glucose levels* while the risk factors are *familial hypercholesterolaemia* and *obesity* and the possible treatment *antidiabetic therapy*. In this case we know that familial hypercholesterolaemia increases the chance for lipid disorder and that lipid disorder in turn causes the blood cholesterol levels to be elevated. Similarly we know that antidiabetic therapy decreases the chance of having diabetes mellitus while having diabetes mellitus increases the blood glucose levels. Obesity is also known to cause both lipid disorder and diabetes mellitus. These are all causal relations and hence can be represented as directed edges in our CG. Finally we also know that there exists a correlation between diabetes mellitus and lipid disorder that cannot be explained only by the common parent obesity. I.e. if a person has diabetes mellitus he or she is more likely to also have lipid disorder than another person that does not have diabetes mellitus, even if they have the same level obesity. This correlation is not causal since it would be wrong to say that diabetes mellitus causes lipid disorder or vice versa and hence we represent the correlation with a non-directed edge. The resulting CG can be seen in Fig. 10.7.

As noted above the process so far corresponds well to that of BNs. The difficulty now is to choose which interpretation to use and thereafter to check that the CG can represent the dependences that exist in the system according to our expert knowledge. In some cases this might be easy and we might identify the non-causal correlation as a relation typically represented by a certain CG interpretation. This can for example be if we know that there exists some hidden common cause between variables that has non-directed edges between them (MVR CG) or if these relations are better described as feedback relations (LWF CG). In many cases we might however not have this information and we are then left to study the represented independence model. The first thing one can consider is whether or not information should "flow" through from parents of a component to all nodes in the component. In our case this would for example be whether familial hypercholesterolaemia increases the probability of having diabetes mellitus, given that no other information is known. If this is the case, then we know that the LWF CG interpretation is the only CG interpretation representing this dependency. If it is not the case, then we will have to consider both AMP and MVR CGs. To see the difference between these interpretations we need three nodes X, Y and Z in the same component such that X is adjacent of Y and Y is adjacent of Z while X and Z are non-adjacent. Then, if $X \perp Z | pa(X)$, we know that the relation is best represented by a MVR CG, while if $X \not\perp Z | pa(X)$, it

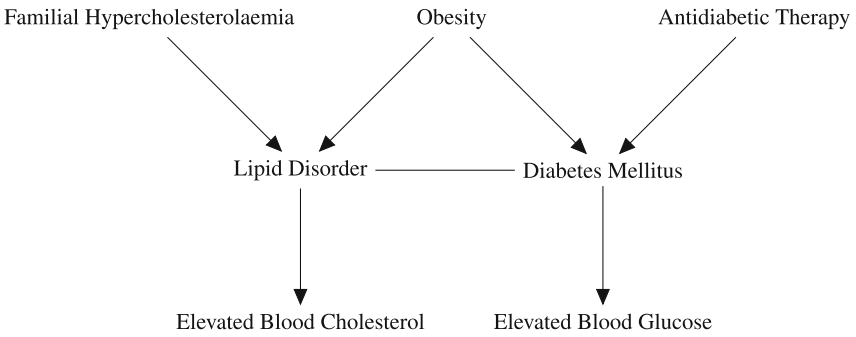


Fig. 10.7 The CG corresponding to the lipid disorder-diabetes mellitus example with a non-directed edge between lipid disorder and diabetes mellitus

is best represented by an AMP CG. Finding the best interpretation becomes even more problematic if multiple types of non-causal relations have been included in the model, corresponding to different CG interpretations. In such a case one either has to choose the interpretation that fits most of the relations or choose an even more general PGM class than CGs. In our example we can note that familial hypercholesterolaemia does in fact increase the probability of having diabetes mellitus, given no other information, and hence we want to use the LWF CG interpretation. This also corresponds well with the authors choice, even though their choice is based on that lipid disorder and diabetes mellitus have a feedback relation between them and that the diseases almost always are in some kind of equilibrium [10].

Once a CG interpretation have been chosen it is also important to make sure that the model can represent all desired (conditional) dependences. If a (conditional) dependency is not represented an extra edge will have to be added. This is of course undesirable since it obscures the "true" relations in the system but as always in PGM modelling we want a model accurately representing all dependences in the underlying system while still representing as many independences as possible. This last step is especially important if the non-causal relation modelled does not perfectly correspond to a CG interpretation or if multiple types of non-causal relations exist in the modelled system.

10.4.2 Learning CGs Under the Faithfulness Assumption

All structure learning algorithms that exist for CGs today are constraint based and assume that the data comes in the form of a probability distribution p. Such a distribution can for example be found through a set of samples of the system. In this subsection we will cover the case where we assume that p is faithful to some CG while we in the next subsection relax that assumption. We say that a probability distribution p is faithful to some CG G iff G have the same separations and non-separations as

independences and dependences in p, i.e. that G can perfectly represent the independence model of p. This means that a probability distribution p that is faithful to some LWF CG G not necessarily is faithful to some AMP CG H, and hence that faithfulness is dependent on the PGM class we have in mind [23].

It is important to stress that this is a strong assumption. However, faithfulness allows for very fast and efficient algorithms since the reasoning in the algorithms can be made in the space of all CG models, instead of in the much larger space of all independence models. Today there exist structure learning algorithms for all three interpretations under the faithfulness assumption. Three of these are based on the PC algorithm [15, 27] used for BNs and contain three phases. In the first phase they learn the adjacencies of the CG; in the second they orient some of the edges according to simple rules; and in the third the remaining edges are oriented to avoid semi-directed cycles. This allows for an efficient way of learning the structure where no step has to be backtracked. For a comprehensive treatment of these algorithms we refer the reader to Studený's work [29] for LWF CGs, Peña's work for AMP CGs [19] and Sonntag and Peña's work [24] for MVR CGs. Finally there also exists a second, decomposition-based algorithm for learning LWF CGs developed by Ma et al. that has been shown to be of lower complexity than the PC variant algorithm [14]. It should be noted that since all structure learning algorithms are constraint based they will only find a CG with the correct independence model. Finding the CG with the correct causal explanation requires additional expert knowledge or experiments. However, having a CG with the correct independence model allows us find all possible causal explanations and their corresponding CGs.

10.4.3 Weakening the Faithfulness Assumption

It has been argued that it is unlikely that a randomly generated probability distribution that factorizes according to a BN is unfaithful to the BN [16]. While this is true if every parameter in a BN is generated randomly, the argument may not hold if the parameters have been hand picked (e.g. by a designer or by nature through evolution). Needless to say these are the systems we are mostly interested in modelling.

If one would apply the learning algorithms described in the previous subsection on a probability distribution that is not faithful to a CG of the appropriate interpretation it can no longer be guaranteed that the learnt CG can factorize the probability distribution properly. This means that the learnt CG might represent independences that do not exist in the underlying system which the probability distribution represents. Hence there might exist relations between variables in the underlying system that are not represented in the CG model. Moreover this means that no matter how the CG is parametrized it can never represent the original probability distribution perfectly. This is of course a problem since we would like to learn an inclusion optimal CG, i.e. a CG that can factorize the probability distribution, but contains as many separations as possible [20].

Unfortunately, learning a CG without assuming faithfulness is very complex and computationally demanding. The only algorithm for this task in the current literature is the CKES algorithm for LWF CGs presented by Peña et al. [20], which is based on a similar algorithm for BNs called KES [17]. The algorithm works by iteratively adding (resp. removing) separations between variables in the CG that are independent (resp. dependent) in the probability distribution given their boundary in the CG of that iteration. This is performed by removing (resp. adding) the appropriate edges in the CG. Moreover, to ensure that an inclusion optimal CG is reached at the end of the algorithm all CGs in the Markov equivalence class of the CG in any iteration may have to be searched for improvements. Like all efficient learning algorithms certain assumptions do however have to be made about the probability distribution. These are that the independence model induced by it fulfills the graphoid properties as well as the composition property [20]. The graphoid properties are satisfied for all strictly positive probability distributions, while the composition property is satisfied for every Gaussian probability distribution.

10.4.4 Factorisation and Parameter Learning

Hitherto very little research has been done on CG parameter learning and hence it is one of the weak points of CGs. Although parametrizations exist for all three CG interpretations for continuous variables [1, 3, 18, 31] it exists no efficient way of learning these parameters from a probability distribution. Instead iterative algorithms have to be used similarly as for MNs. We will here show an example of how this is done for LWF CGs.

The factorisation of a probability distribution p with variables $X_1, \ldots, X_n$ according to a LWF CG G with components $K_1, \ldots, K_m$ is

$$p(X_1, ..., X_n) = \prod_{i=1}^{m} p(K_i | pa_G(K_i)). \tag{10.1}$$

Each component K_i can then be factorized clique-wise as follows

$$p(K_i | pa_G(K_i)) = \frac{1}{Z_i} \prod_{M \in M_C} \phi_M, \tag{10.2}$$

where M_C are the complete subsets in the closure graph of K_i, i.e. the induced subgraph $G_{K_i \cup pa_G(K_i)}$ where each directed edge is replaced by an undirected edge and each pair of vertices in $pa_G(K_i)$ also are connected by an undirected edge. Each ϕ_M is then a potential over the variables in M and Z_i is a normalization constant. In other words, the probability distribution of the closure graph of each component can be seen as a MN. To parametrize these products and potentials we can then simply parametrize the system of linear equations since there exists a one to one relation between it and the probability distribution.

Another way to parametrize LWF CGs have been introduced by Lappenschaar et al. [10]. They proposed a qualitative approach to LWF CGs in which it is only calculated whether two variables adjacent in the graph have positive, negative or ambiguous influence on each other, and not the actual parameter value. In the article Lappenschaar et al. describes how these parameters can be learnt from data and uses the approach for modelling the interaction between diabetes and lipid disorder given the relevant factors. Their results show that one of the advantages of using qualitative LWF CGs compared to qualitative BNs is the ability to capture equilibrium models.

10.5 Summary

In this chapter we have shown how CGs can be used to model complex system such as gene networks. We have also shown some advantages of using CGs compared to using BNs, MNs or covGs, which are more commonly used in real-world applications today. The main advantage is that CGs are more flexible since they can represent both causal and non-causal relations and thereby represent a larger set of independence models compared to BNs, MNs or covGs. This means that CGs can express a model that is closer, or at least as close, to the real system as any BN, MN or covG. At the same time, they are still easy to interpret and one can relate their structure to the underlying molecular processes.

We have also discussed structure learning algorithms for all of the CG interpretations. Using these algorithms on samples from an advanced system like a gene network will result in a CG which may give good insight into how the variables in the system interact, even if it contains non-causal relations between its variables.

10.6 Appendix, System of Linear Equations for CGs

In this appendix we derive and present how the separation criteria of the different interpretations translate into systems of linear equations.

10.6.1 LWF CGs

Let G be a LWF CG with connectivity components $K_1, \ldots, K_n$. Let $\mathcal{N}(G)$ denote the set of regular Gaussian distributions that factorize with respect to G, which coincide with the set of distributions that satisfy the LWF global Markov property with respect to G [11, Theorems 3.34 and 3.36]. Let $p \in \mathcal{N}(G)$. Assume without loss of generality that p has mean 0. Let $\Omega^i_{K_i,K_i}$ and $\Omega^i_{K_i,pa_G(K_i)}$ denote submatrices of the precision matrix Ω^i of $p(K_i, pa_G(K_i))$. Then, as shown in [2, Sect. 2.3.1],

$$K_i|pa_G(K_i) \sim \mathcal{N}(\beta^i\, pa_G(K_i),\, \Lambda^i) \tag{10.3}$$

where

$$\beta^i = -(\Omega^i_{K_i,K_i})^{-1}\Omega^i_{K_i,pa_G(K_i)} \tag{10.4}$$

and

$$(\Lambda^i)^{-1} = \Omega^i_{K_i,K_i}. \tag{10.5}$$

Then, as shown in [18, Sect. 3], G has associated a system of linear equations with normally distributed errors as follows. For every K_i,

$$K_i = \beta^i\ pa_G(K_i) + \epsilon^i \tag{10.6}$$

where

$$\epsilon^i \sim \mathcal{N}(0,\, \Lambda^i) \tag{10.7}$$

and

$$(\Omega^i_{K_i,K_i})_{j,k} = 0 \text{ for all } j,k \in K_i \text{ such that } j-k \text{ is not in } G \tag{10.8}$$

and

$$(\Omega^i_{K_i,pa_G(K_i)})_{j,k} = 0 \text{ for all } j \in K_i \text{ and } k \in pa_G(K_i) \text{ such that } j \leftarrow k \text{ is not in } G. \tag{10.9}$$

It is worth mentioning that the mapping above between the probability distributions in $\mathcal{N}(G)$ and the systems of linear equations is bijective [18, Lemma 1]. Moreover, an alternative (but equivalent) parameterization of the probability distributions in $\mathcal{N}(G)$ is presented in [30].

Then, G has associated a system of linear equations with correlated errors as follows. For every $X_j \in K_i$,

$$X_j = \beta_j\ pa_G(K_i) + \epsilon^j \tag{10.10}$$

where

$$\beta_j \text{ is the } j- \text{ th row of } \beta^i \tag{10.11}$$

and

$$Cov(\epsilon^j, \epsilon^k) = (\Lambda^i)_{j,k}. \tag{10.12}$$

Note that X_j is a linear combination of $pa_G(K_i)$ and not of $pa_G(X_j)$. Note also that, as shown in [32, Propositon 5.7.3],

$$(\Omega^i_{K_i,K_i})^{-1} = \Sigma_{K_i \cdot pa_G(K_i)} \tag{10.13}$$

where $\Sigma_{K_i \cdot pa_G(K_i)}$ represents the partial covariance matrix of K_i given $pa_G(K_i)$.

Then, as shown in [8, Theorem 1], the element (A, B) of $(\Omega^i_{K_i,K_i})^{-1}$ can be written as a sum of path weights over all the paths in G between A and B through nodes in K_i. Specifically,

$$((\Omega^i_{K_i,K_i})^{-1})_{A,B} = (\Sigma_{K_i \cdot pa_G(K_i)})_{A,B} = \sum_{\rho \in \rho_{A,B}} (-1)^{|\rho|+1} \frac{|(\Omega^i_{K_i,K_i})_{\setminus\rho}|}{|\Omega^i_{K_i,K_i}|} \prod_{l=1}^{|\rho|-1} (\Omega^i_{K_i,K_i})_{\rho_l,\rho_{l+1}} \tag{10.14}$$

where $\rho_{A,B}$ denotes the set of paths in G between A and B through nodes in K_i, $|\rho|$ denotes the number of nodes in a path ρ, ρ_l denotes the l-th node in ρ, and $(\Omega^i_{K_i,K_i})_{\setminus\rho}$ is the matrix with the rows and columns corresponding to the nodes in ρ omitted. Moreover, the determinant of a zero-dimensional matrix is taken to be 1. This leads to the following interpretation of β_j: By Eqs. 10.4, 10.11 and 10.14, the k-th element of β_j can be written as sum of path weights over all the paths in G between X_k and X_j trough nodes in K_i.

10.6.2 AMP CGs

Let G be an AMP CG with connectivity components $K_1, \ldots, K_n$. Let $\mathcal{N}(G)$ denote the set of regular Gaussian distributions that satisfy the AMP global Markov property with respect to G. Let $p \in \mathcal{N}(G)$. Assume without loss of generality that p has mean 0. Then, as shown above, $K_i | pa_G(K_i) \sim \mathcal{N}(\beta^i pa_G(K_i), \Lambda^i)$. Then, as shown in [1, Sect. 5], G has associated a system of linear equations with normally distributed errors as follows. For every K_i,

$$K_i = \beta^i \; pa_G(K_i) + \epsilon^i \tag{10.15}$$

where

$$\epsilon^i \sim \mathcal{N}(0, \Lambda^i) \tag{10.16}$$

and

$$((\Lambda^i)^{-1})_{j,k} = 0 \text{ for all } j, k \in K_i \text{ such that } j - k \text{ is not in } G \tag{10.17}$$

and

$$(\beta^i)_{j,k} = 0 \text{ for all } j \in K_i \text{ and } k \in pa_G(K_i) \text{ such that } j \leftarrow k \text{ is not in } G. \tag{10.18}$$

It is worth mentioning that the mapping above between the probability distributions in $\mathcal{N}(G)$ and the systems of linear equations is bijective [1, Sect. 5]. Moreover, the first constraint here coincides with the first constraint in the previous section.

Then, G has associated a system of linear equations with correlated errors as follows. For every $X_j \in K_i$,

$$X_j = \beta_j \; pa_G(X_j) + \epsilon^j \tag{10.19}$$

where

$$\beta_j \text{ contains the nonzero elements of } (\beta^i)_j. \tag{10.20}$$

and

$$Cov(\epsilon^j, \epsilon^k) = (\Lambda^i)_{j,k}. \tag{10.21}$$

Note that, unlike in the previous section, X_j is here a linear combination of $pa_G(X_j)$ and not of $pa_G(K_i)$.

10.6.3 MVR CGs

Let G be a MVR CG with connectivity components $K_1, \ldots, K_n$. Then, G has associated a system of linear equations with normally distributed errors as shown in the previous section except for two differences. First, $\mathcal{N}(G)$ now denotes the set of regular Gaussian distributions that satisfy the MVR global Markov property with respect to G. Second, we now replace Eq. 10.17 with

$$(\Lambda^i)_{j,k} = 0 \text{ for all } j, k \in K_i \text{ such that } j \leftrightarrow k \text{ is not in } G. \tag{10.22}$$

See also [9].

Acknowledgements. This work is funded by the Center for Industrial Information Technology (CENIIT) and a so-called career contract at Linköping University, and by the Swedish Research Council (ref. 2010-4808).

References

1. Andersson, S.A., Madigan, D., Perlman, M.D.: An alternative Markov property for chain graphs. Scand. J. Stat. **28**, 33–85 (2001)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Cox, D.R., Wermuth, N.: Linear dependencies represented by chain graphs. Stat. Sci. **8**, 204–218 (1993)
4. Cox, D.R., Wermuth, N.: Multivariate Dependencies: Models, Analysis and Interpretation. Chapman and Hall, London (1996)
5. Drton, M.: Discrete chain graph models. Bernoulli **15**, 736–753 (2009)
6. Ferráandiz, J., Castillo, E.F., Snamartín, P.: Temporal aggregation in chain graph models. J. Stat. Plann. Infer. **133**, 69–93 (2005)
7. Frydenberg, M.: The chain graph Markov property. Scand. J. Stat. **17**, 333–353 (1990)
8. Jones, B., West, M.: Covariance decomposition in undirected Gaussian graphical models. Biometrika **92**, 779–786 (2005)
9. Kang, C., Tian, J.: Markov properties for linear causal models with correlated errors. J. Mach. Learn. Res. **10**, 41–70 (2009)

10. Lappenschaar, M., Hommersom, A., Lucas, P.J.F.: Qualatitive chain graphs and their application. Int. J. Approximate Reasoning **55**, 957–976 (2014)
11. Lauritzen, S.L.: Graphical Models. Clarendon Press, Oxford (1996)
12. Lauritzen, S.L., Richardson, T.S.: Chain graph models and their causal interpretations. J. Roy. Stat. Soc. B **64**, 321–361 (2002)
13. Lauritzen, S.L., Wermuth, N.: Graphical models for association between variables, some of which are qualitative and some quantitative. Ann. Stat. **17**, 31–57 (1989)
14. Ma, Z., Xie, X., Geng, Z.: Structural learning of chain graphs via decomposition. J. Mach. Learn. Res. **9**, 2847–2880 (2008)
15. Meek, C.: Causal inference and causal explanation with background knowledge. In: Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 403–410 (1995)
16. Meek, C.: Strong completeness and faithfulness in Bayesian networks. In: Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 411–418 (1995)
17. Nielsen, J.D., Kočka, T., Peña, J.M.: On local optima in learning Bayesian networks. In: Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence, pp. 435–442 (2003)
18. Peña, J.M.: Faithfulness in chain graphs: the Gaussian case. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, pp. 588–599 (2011)
19. Peña, J.M.: Learning AMP chain graphs under faithfulness. In: Proceedings of the 6th European Workshop on Probabilistic Graphical Models, pp. 251–258 (2012)
20. Peña, J.M., Sonntag, D., Nielsen, J.: An inclusion optimal algorithm for chain graph structure learning. In: Proceedings of the 17th International Conference on Artificial Intelligence and Statistics, pp. 778–786 (2014)
21. Richardson, T.S.: Markov properties for acyclic directed mixed graphs. Scand. J. Stat. **30**, 145–157 (2003)
22. Segal, E., et al.: Learning module networks. J. Mach. Learn. Res. **6**, 557–588 (2005)
23. Sonntag, D., Peña, J.M.: Chain graph interpretations and their relations. In: van der Gaag, L.C. (ed.) ECSQARU 2013. LNCS, vol. 7958, pp. 510–521. Springer, Heidelberg (2013)
24. Sonntag, D., Peña, J.M.: Learning multivariate regression chain graphs under faithfulness. In: Proceedings of the 6th European Workshop on Probabilistic Graphical Models, pp. 299–306 (2012)
25. Sonntag, D.: On expressiveness of the AMP chain graph interpretation. In: van der Gaag, L.C., Feelders, A.J. (eds.) PGM 2014. LNCS, vol. 8754, pp. 458–470. Springer, Heidelberg (2014)
26. Sonntag, D., Peñna, J.M., Gómez-Olmedo, M.: Approximate counting of graphical models via MCMC revisited. Int. J. Intell. Syst. **30**, 384–420 (2015)
27. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. Springer, New York (1993)
28. Studený, M.: Bayesian networks from the point of view of chain graphs. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 496–503 (1998)
29. Studený, M.: On recovery algorithms for chain graphs. Int. J. Approximate Reasoning **17**, 265–293 (1997)
30. Wermuth, N.: On block-recursive linear regression equations (with discussion). Braz. J. Probab. Stat. **6**, 1–56 (1992)
31. Wermuth, N., Wiedenbeck, M., Cox, D.R.: Partial inversion for linear systems and partial closure of independence graphs. BIT Numer. Math. **46**, 883–901 (2006)
32. Whittaker, J.: Graphical Models in Applied Multivariate Statistics. Wiley, New York (1990)