



## Independencies Induced from a Graphical Markov Model After Marginalization and Conditioning: The R Package `ggm`

Giovanni M. Marchetti  
University of Florence

---

### Abstract

We describe some functions in the R package `ggm` to derive from a given Markov model, represented by a directed acyclic graph, different types of graphs induced after marginalizing over and conditioning on some of the variables. The package has a few basic functions that find the essential graph, the induced concentration and covariance graphs, and several types of chain graphs implied by the directed acyclic graph (DAG) after grouping and reordering the variables. These functions can be useful to explore the impact of latent variables or of selection effects on a chosen data generating model.

*Keywords:* conditional independence, directed acyclic graphs, covariance graphs, concentration graphs, chain graphs, multivariate regression graphs, latent variables, selection effects.

---

## 1. Introduction

The R package `ggm` (see [Marchetti and Drton \(2006\)](#)) has been designed primarily for fitting Gaussian Graphical Markov models based on directed acyclic graphs (DAGs), concentration and covariance graphs and ancestral graphs. The package is a contribution to the gR-project described in [Lauritzen \(2002\)](#).

Special functions are designed to define easily some basic types of graphs by using model formulae. For example, in [Figure 1](#) the directed acyclic graph to the right is defined with the function `DAG` shown top left and its output is the adjacency matrix of the graph.

In the package `ggm` all graph objects are defined simply by their adjacency matrix with special codes for undirected, directed, or bidirected edges. In the future, `ggm` should be extended to comply to the classes and methods developed by [Højsgaard and Dethlefsen \(2005\)](#) in the `gRbase` package.

In this paper we describe some functions in the `ggm` package which are concerned with graph

```

> dag = DAG(Y~X+U, X~Z+U)
> dag
  Y X U Z
Y 0 0 0 0
X 1 0 0 0
U 1 1 0 0
Z 0 1 0 0

```

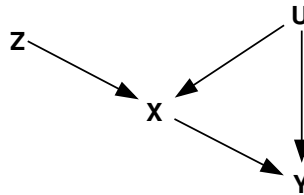


Figure 1: A generating directed acyclic graph obtained from the function `DAG` by using model formulae.

computations. In particular, we describe procedures to derive from the adjacency matrix of a generating DAG and an ordered partition of its vertices, an implied chain graph after grouping and reordering the variables as specified by the partition. The theory is developed in [Wermuth and Cox \(2004\)](#). Our focus however is on the details of the implementation in R and on examples of application. This paper is a tutorial on available R software tools to investigate the effects on a postulated generating model of latent variables or of selection effects.

In Section 2 we introduce the kind of graphs discussed (mixed graphs), the conventions used to specify their adjacency matrices and the syntax of model formulae for directed acyclic graphs. This syntax is especially suggestive of associated linear recursive structural models. Then we describe some basic functions for d-separation tests and for obtaining classes of Markov equivalent directed acyclic graphs. In Section 3 functions for deriving undirected graphs implied by a generating DAG are discussed and some details on the algorithms used in their computation are given in the appendix. Chain graphs with different interpretations, compatible with the generating graph are explained in Section 4.

## 2. Defining graphs

### 2.1. Graphs and adjacency matrices

A graph  $G = (V, \mathcal{E})$  is defined by a set of vertices  $V$  (also called nodes) and a set of edges  $\mathcal{E}$ . The edges considered in package `ggm` may be *directed* ( $\rightarrow$ ), *undirected* ( $—$ ) or *bidirected* ( $\leftrightarrow$ ). Here we will use dashed bidirected edges  $\leftrightarrow$  as a compromise between the notation by [Cox and Wermuth \(1996\)](#) which use dashed undirected edges and that by [Richardson and Spirtes \(2002\)](#), which use solid bidirected edges  $\longleftrightarrow$ . For a similar notation see also [Pearl \(2000\)](#).

Graphs considered in this paper may be represented by a square matrix  $E = [e_{ij}]$ , for  $(i, j) \in V \times V$ , called *adjacency matrix* where  $e_{ij}$  can be either 0, 1, 2 or 3. The basic rules of these adjacency matrices are as follows.

- We always assume that  $e_{ii} = 0$  for any node  $i \in V$ .
- If there is no edge between two distinct nodes  $i$  and  $j$ , then  $e_{ij} = e_{ji} = 0$  and we say that  $i$  and  $j$  are not adjacent. These zero values in the adjacency matrix are also called *structural zeros*.

- A directed edge  $i \leftarrow j$  is represented by  $e_{ij} = 0$  and  $e_{ji} = 1$ .
- An undirected edge  $i \text{---} j$  is represented by  $e_{ij} = 1$  and  $e_{ji} = 1$ .
- A bidirected edge  $i \leftrightarrow j$  is represented by  $e_{ij} = 2$  and  $e_{ji} = 2$ .
- A *double edge*  $i \overleftrightarrow{\leftarrow} j$  is represented by  $e_{ij} = 2$  and  $e_{ji} = 3$ . This is the only allowed situation with multiple edges between two nodes. This case is called *bow pattern* by Brito and Pearl (2002).

If a graph contains only undirected edges it is sometimes called a *concentration graph*, if it contains only bidirected edges it is called a *covariance graph*. The adjacency matrices of covariance and concentration graphs are symmetric. If a graph contains only directed edges it is a *directed graph*. If it does not contain cycles (a cycle is a direction preserving path with identical end points) it is called *directed acyclic graph* (DAG).

Wermuth and Cox (2004) define an equivalent representation of graphs based on a binary matrix called *edge matrix*. The edge matrix  $\mathcal{M}$  of a graph is related to its adjacency matrix  $E$  by  $\mathcal{M} = E^\top + I$ , where  $I$  is the identity matrix of the same size as  $E$  and  $^\top$  denotes transposition.

## 2.2. Model formulae for DAGs

In package **ggm** there are no special objects of class **graph** and graphs are represented simply by adjacency matrices with rows and columns labeled with the names of the vertices. The graphs can be plotted using the function **drawGraph** which is a minimal but useful tool for displaying small graphs with a rudimentary interface for adjusting nodes and edges. A more powerful interface is provided by the R package **dynamicGraph** (Badsberg 2005).

Usually, we don't want to define graphs by specifying directly the adjacency matrix, but we prefer a more convenient way through constructor functions. We will often use the function **DAG** that defines a directed acyclic graph from a set of regression model formulae. Each formula indicates simply the response variables and the direct influences, i.e. a node  $i$  and the set of its *parents*  $\text{pa}(i) = \{j \in V : i \leftarrow j\}$ . In the formula, the node and its parents are written as in R linear regression models.

*Example 1* The graph in Figure 1 is defined by the command **DAG**( $Y \sim X+U$ ,  $X \sim Z+U$ ) which essentially is a set of symbolic *linear structural equations*. This graph is indeed the representation of the independencies implied by the linear recursive equations

$$\begin{aligned} Y &= \alpha X + \beta U + \epsilon_Y \\ X &= \gamma Z + \delta U + \epsilon_X \\ U &= \epsilon_U \\ Z &= \epsilon_Z \end{aligned} \tag{1}$$

with independent residuals  $\epsilon$ . Note that we omitted the formulae for nodes associated to exogenous variables  $Z$  and  $U$ .

The linear structural equations can be rewritten in reduced form

$$\begin{pmatrix} 1 & -\alpha & -\beta & 0 \\ 0 & 1 & -\gamma & -\delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Y \\ X \\ U \\ Z \end{pmatrix} = \begin{pmatrix} \epsilon_Y \\ \epsilon_X \\ \epsilon_U \\ \epsilon_Z \end{pmatrix}$$

or  $AV = \epsilon$  where  $V = (Y, X, U, Z)^\top$  and  $A$  is an upper triangular matrix. The edge matrix  $\mathcal{A}$  associated with the DAG is a binary matrix obtained from  $A$  by replacing each non zero element by 1. This is obtained with the function `edgeMatrix`:

```
> edgeMatrix(dag)
  Y X U Z
Y 1 1 1 0
X 0 1 1 1
U 0 0 1 0
Z 0 0 0 1
```

Edge matrices play an important role in the algorithms used in **ggm** to compute the induced graphs (see the appendix).

### 2.3. DAG models and the Markov property

Let us denote by  $V$  the set of nodes and by  $E = [e_{ij}]$  the adjacency matrix of a directed acyclic graph  $D = (V, E)$ . Now we suppose that a set of random variables  $Y = (Y_i), i \in V$  can be associated to the nodes  $V$  of the graph such that their density  $f_Y$  factorizes as

$$f_Y(y) = \prod_{i \in V} f(y_i | y_{\text{pa}(i)})$$

If this is the case then the random vector  $Y$  is said to be Markov with respect to the DAG  $D$ . Often the DAG together with a compatible ordering of the nodes is thought as representing a process by which the data are generated. This generating graph is sometimes called the *parent graph* (Wermuth and Cox 2004).

Then, to the set of missing edges of the DAG (or equivalently to the set  $M = \{(i, j) : e_{ij} = 0, e_{ji} = 0\}$  of structural zeros of the adjacency matrix) corresponds a set of conditional independencies that are satisfied by all distributions that are Markov with respect to the graph. These independencies are encoded in the so called Markov properties of the DAG, (see Lauritzen (1996)) and can be read off the graph using the concept of d-separation, Pearl (1988). Two nodes  $i$  and  $j$  are d-separated given a set  $C$  not containing  $i$  or  $j$  if there exist no path between  $i$  and  $j$  such that every collision node  $\rightarrow \circ \leftarrow$  on the path has a descendant in  $C$  and no other node on the path is in  $C$ . Two disjoint sets of nodes  $A$  and  $B$  are d-separated given a third disjoint set  $C$  if for every node  $i \in A$  and  $j \in B$ ,  $i$  and  $j$  are d-separated given  $C$ . Then, for the global Markov property two disjoint sets  $A$  and  $B$  of variables are independent given a third disjoint set  $C$ ,  $A \perp\!\!\!\perp B | C$ , if  $A$  and  $B$  are d-separated by  $C$  in the DAG. We can verify if two sets of nodes  $A$  and  $B$  are d-separated given  $C$  with the function `dSep`.

*Example 2* It is immediate to check that in the DAG of Figure 1 nodes  $Z$  and  $U$  are not d-separated given  $Y$

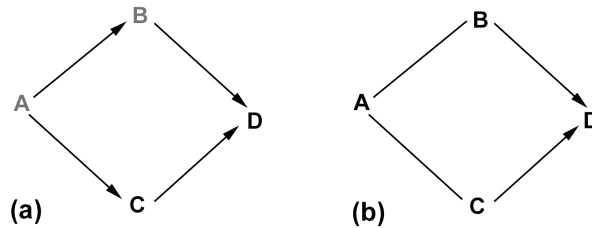


Figure 2: A DAG, in panel (a), and the associated essential graph, in panel (b).

```
> dSep(dag, first="Z", second="U", cond="Y")
[1] FALSE
> dSep(dag, first="Z", second="U", cond=NULL)
[1] TRUE
```

while they are d-separated given the empty set.

It is well-known that different DAGs may determine the same statistical model, i.e. the same set of conditional independence restrictions among the variables associated to the nodes. If there are two DAGs  $D_1$  and  $D_2$  and the class of probability measures Markov to  $D_1$  coincide with the class of probability measures Markov to  $D_2$ , then  $D_1$  and  $D_2$  are said *Markov-equivalent*. Verma and Pearl (1990) and Frydenberg (1990) have shown that two DAGs are Markov equivalent iff they have the same nodes, the same set of adjacencies and the same immoralities (i.e. configurations  $i \rightarrow j \leftarrow k$  with  $i$  and  $k$  not adjacent). The class of all DAGs Markov-equivalent to a given DAG  $D$  can be summarized by the *essential graph*  $D^*$  associated with  $D$ . The essential graph has the same set of adjacencies of  $D$ , but there is an arrow  $i \rightarrow j$  in  $D^*$  if and only if there is an arrow  $i \rightarrow j$  in every  $D'$  belonging to the class of all DAGs Markov-equivalent to  $D$ . In **ggm** the essential graph associated to  $D$  is computed by the function `essentialGraph`.

*Example 3* For instance let us define a DAG with  $D = \text{DAG}(D \sim B + C, B \sim A, C \sim A)$ , see Figure 2(a). Then the essential graph is

```
> essentialGraph(D)
  U Y Z X
U 0 0 0 0
Y 1 0 0 1
Z 1 0 0 1
X 0 1 1 0
```

see Figure 2(b). It has two essential arrows  $Z \rightarrow U \leftarrow Z$  but the arrows  $X \rightarrow Z$  and  $X \rightarrow Y$  can be reversed (but not simultaneously) without changing the set of independencies.

### 3. Undirected graphs induced from a generating DAG

In the following we are interested in studying the independencies entailed by a DAG which is assumed to describe the data generating process. These independencies can be encoded in

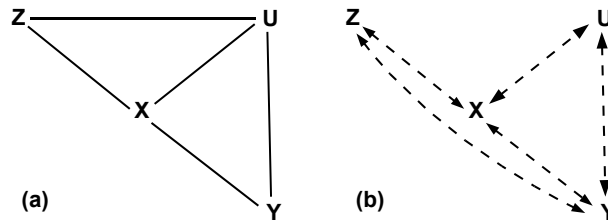


Figure 3: The overall induced concentration (a)) and covariance graphs (b) for the parent graph of Figure 1.

several types of "induced" graphs, and the rest of the paper describes the basic functions of package `ggm` that can be used to derive them.

### 3.1. Overall concentration and covariance graph

The simplest graph induced by a DAG is the *overall concentration graph* which has an undirected edge  $i - j$  iff the DAG does not imply  $i \perp\!\!\!\perp j | V \setminus \{i, j\}$ . Thus the concentration graph may be interpreted via the pairwise Markov property of undirected graphs. The overall concentration graph is obtained after the operation of *moralization* of the DAG. This means that if two nodes  $i$  and  $j$  are adjacent in the DAG then they are adjacent in the concentration graph, and if they are not adjacent but have a common response,  $i \rightarrow \circ \leftarrow j$  in the DAG, then an additional edge  $i - j$  results in the concentration graph.

Another important induced graph is the *overall covariance graph*. This is a graph with all bidirected (dashed) edges which has an edge  $i \leftrightarrow j$  iff  $D$  does not imply  $i \perp\!\!\!\perp j$ . In this graph  $i$  and  $j$  are not adjacent if there are only collision paths between  $i$  and  $j$  in the parent graph. The covariance graph is also a simple example of an *ancestral graph* (Richardson and Spirtes 2002).

The adjacency matrix of the overall concentration and of the covariance graph are computed by the functions `inducedConGraph` and `inducedCovGraph`.

*Example 4* From the DAG in Figure 1 we compute the covariance and concentration graphs shown in Figure 3 as follows

```
> inducedConGraph(dag)
  Y X U Z
Y 0 1 1 0
X 1 0 1 1
U 1 1 0 1
Z 0 1 1 0
> inducedCovGraph(dag)
  Y X U Z
Y 0 2 2 2
X 2 0 2 2
U 2 2 0 0
Z 2 2 0 0
```

Note that from the triangular system of linear structural equations (1) associated with the DAG (assuming for simplicity that the all the residuals have unit variances) the concentration matrix of the variables is

$$\Sigma^{-1} = A^{\top} A = \begin{pmatrix} 1 & -\alpha & -\beta & 0 \\ \cdot & 1 + \alpha^2 & \alpha\beta - \gamma & -\delta \\ \cdot & \cdot & 1 + \beta^2 + \gamma^2 & \gamma\delta \\ \cdot & \cdot & \cdot & 1 + \delta^2 \end{pmatrix}$$

where the dot notation indicates here that in a symmetric matrix the  $ji$ -element coincides with the  $ij$ -element. Thus the concentration  $\sigma^{YZ}$  is zero for every choice of the parameters of the linear structural equations and this structural zero coincides with that of the adjacency matrix of the concentration graph of Figure 3(a). The remaining concentrations are in general not zero. The correspondence between structural zeros in the parameter matrices derived from the linear recursive equations associated to a DAG and the zero entries in the adjacency matrix of the induced graphs has been exploited by [Wermuth and Cox \(2004\)](#) to obtain explicit matrix expressions for the adjacency matrices of the induced graphs (see appendix).

### 3.2. Concentration and covariance graphs after conditioning and marginalization

The above definitions of induced overall concentration and covariance graph can be extended by considering two disjoint subsets of nodes  $C$  and  $M$  of  $V$  and by looking at the independence structures in the system of the remaining variables  $S = V \setminus (C \cup M)$  ignoring the variables in  $M$  and conditional on variables in  $C$ .

[Cox and Wermuth \(1996\)](#) define:

- $\mathcal{S}^{SS|C}$ : the induced concentration graph of  $S$  given  $C$ , which has an edge  $i - j$ , for  $i$  and  $j$  in  $S$ , iff the DAG does not imply  $i \perp\!\!\!\perp j \mid C \cup S \setminus \{i, j\}$ .
- $\mathcal{S}_{SS|C}$ : the induced covariance graph of  $S$  given  $C$ , which has an edge  $i \leftrightarrow j$ , for  $i$  and  $j$  in  $S$ , iff the DAG does not imply  $i \perp\!\!\!\perp j \mid C$ .

To derive these graphs the R functions `inducedConGraph` and `inducedCovGraph` need the additional arguments `sel` and `cond` to specify the sets  $S$  and  $C$ , respectively.

*Example 5* Let us consider the DAG in Figure 4(a) and suppose that it represents the true data generating process, but that we cannot observe variable  $U$ . Then the covariance graph marginalizing over  $U$  is obtained by the following commands:

```
> dag2 = DAG(Y ~ X+U, W ~ Z+U)
> inducedCovGraph(dag2,
  sel=c("X", "Y", "Z", "W"))
  X Y Z W
X 0 2 0 0
Y 2 0 0 2
Z 0 0 0 2
W 0 2 2 0
```

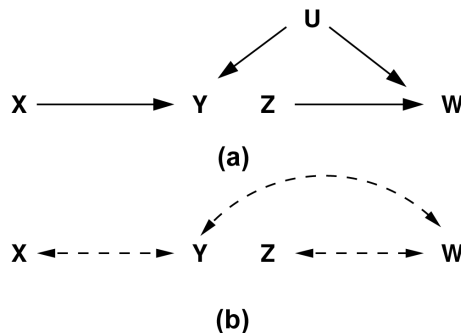


Figure 4: A generating directed acyclic graph (a) and the covariance graph (b) after marginalizing over  $U$ .

see Figure 4 (b). Note that its adjacency matrix is obtained simply by deleting the row and column associated to variable  $U$  from the overall covariance graph of Figure 3 (b). In general, the covariance graph marginalizing over a set of nodes is the subgraph induced by these nodes. We can find analogously the induced concentration graph for variables  $X$ ,  $Y$ ,  $Z$  and  $W$ , marginalizing over  $U$  and this turns out to be complete. In this case the adjacency matrix cannot be obtained simply by deleting one row and one column from the overall concentration graph.

## 4. Induced chain graphs

### 4.1. Induced regression graphs

Other types of induced graphs can be obtained by considering *multivariate regression graphs*. These are graphs where the node set can be partitioned into two subsets  $S$  and  $C$  called blocks such that  $S$  contains the response variables and  $C$  the explanatory variables. Given a generating DAG, the *induced multivariate regression graph* of  $S$  given  $C$ , denoted by  $\mathcal{P}_{S|C}$ , has an arrow  $i \rightarrow j$ , with  $i \in C$  and  $j \in S$ , iff the generating DAG does not imply  $i \perp\!\!\!\perp j \mid C$  (see Cox and Wermuth (1996)). The related **ggm** function is called `inducedRegGraph`. Note that Cox and Wermuth (1996) use dashed arrows for this directed graph, while in **ggm** we use solid arrows.

*Example 6* For the DAG in Figure 4 (a) let us consider  $S = \{W\}$  and  $C = \{X, Y, Z\}$ . Then the regression graph of  $S$  given  $C$  is shown in Figure 5. The function `inducedRegGraph` returns a  $|C| \times |S|$  binary matrix with elements  $e_{ij}$  equal to 1 if the induced regression graph has an arrow  $i \rightarrow j$  and zero otherwise. Note that in this example  $X$  and  $Y$  are both parents of  $W$  in the induced graph even if they are not directly explanatory of  $W$  in the parent graph. This happens because they are connected by a collisionless path to  $W$  in the parent graph. We added to the regression graph the induced covariance graph for the explanatory variables from which we see that  $Z$  is marginally independent of  $(X, Y)$ .

By combining several univariate regression graphs in a specified order we can obtain from the generating graph an induced DAG in the new chosen ordering of the nodes, possibly after



```

> inducedRegGraph(dag2, sel="W",
  cond=c("Y", "X", "Z"))
  W
Y 1
X 1
Z 1
> inducedCovGraph(dag2,
  sel=c("Y", "X", "Z"))
  Y X Z
Y 0 2 0
X 2 0 0
Z 0 0 0

```

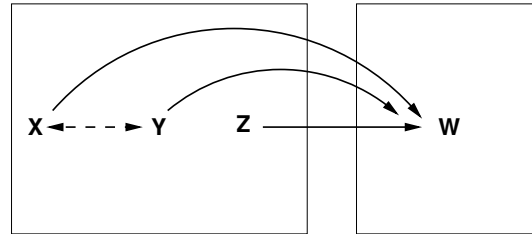


Figure 5: The regression graph induced from the DAG of Figure 4 of  $S = W$  given  $C = \{X, Y, Z\}$ , after marginalizing over  $U$ . Within the block of explanatory variables is drawn the induced covariance graph. On the left: the R code used.

marginalizing and conditioning on some nodes. The relevant function is called `inducedDAG`.

*Example 7* Suppose that we cannot measure variable  $U$  and that the generating process is again specified by the graph of Figure 4 (a). Assume also that the ordering of the variables is known and equal to  $(X, Y, Z, W)$  (from left to right). Then, considering the regression graphs of each variable in turn given all the preceding ones in the ordering, we obtain the DAG shown in Figure 6. Note that this DAG fails to represent the independence relations of the generating graph of Figure 4 (a) (cf. Richardson and Spirtes (2002)).

The previous examples considered only univariate responses, but often we would like to regard some responses jointly.

*Example 8* Consider again the DAG of Figure 4 (a), but with a selected set  $S = \{Y, W\}$  of joint responses and a set of explanatory variables  $C = \{X, Z\}$ . The multivariate regression graph  $\mathcal{P}_{S|C}$  together with a conditional concentration graph  $\mathcal{S}^{SS|C}$  is shown in Figure 7 and a marginal covariance graph  $\mathcal{S}_{CC}$  for the block of explanatory variables  $X$  and  $Z$ , which turn out to be independent. This graph is a *chain graph*, with the so called AMP Markov property interpretation (from Andersson, Madigan, and Perlman (2001)). That is, the implied independencies are  $Y \perp\!\!\!\perp Z|X$ ,  $W \perp\!\!\!\perp X|Z$  and  $X \perp\!\!\!\perp Z$ , which in turn imply  $X \perp\!\!\!\perp (W, Z)$ .

```

> inducedDAG(dag2, order=c("X","Y","Z","W"))
  X Y Z W
X 0 1 0 1
Y 0 0 0 1
Z 0 0 0 1
W 0 0 0 0

```

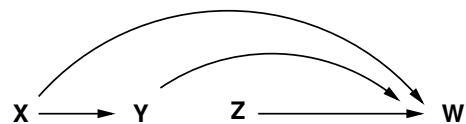


Figure 6: DAG induced from the generating graph of Figure 4 in the ordering  $(X, Y, Z, W)$ , after marginalizing over  $U$ .

```

> inducedRegGraph(dag2, sel=c("Y", "W"),
                  cond=c("X", "Z"))
  Y W
X 1 0
Z 0 1
> inducedCovGraph(dag2, sel=c("Y", "W"),
                  cond=c("X", "Z"))
  Y W
Y 1 0
W 0 1

```

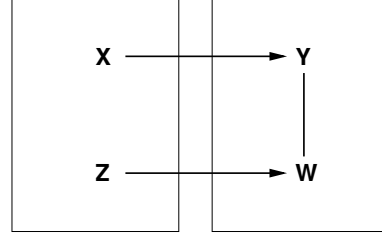


Figure 7: Multivariate regression graph induced from the parent graph of Figure 4, for selected responses  $S = \{Y, W\}$ , conditioning variables  $C = \{X, Z\}$ , after marginalizing over  $U$ . The graph within the block  $S$  of responses is the concentration graph  $\mathcal{S}^{SS|C}$ , while the graph within the block of explanatory variables is a covariance graph  $\mathcal{S}_{CC}$ .

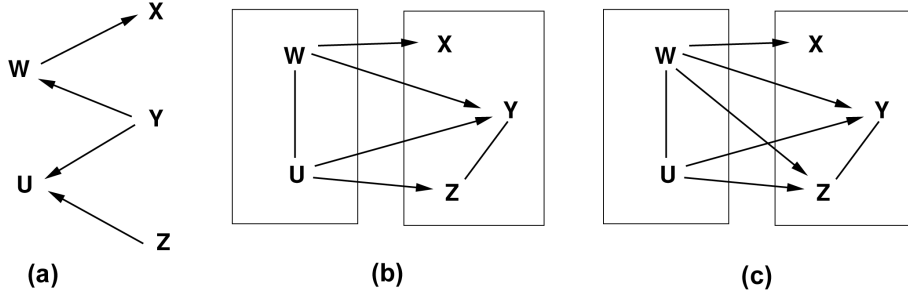


Figure 8: (a) A DAG and (b) the induced chain graph for selected responses  $S = \{X, Y, Z\}$ , and conditioning variables  $C = \{W, U\}$  — the implied independencies follow from the LWF Markov property. (c) A chain graph for the same sets of nodes  $S$  and  $C$ , but with the AMP interpretation.

## 4.2. Different types of induced chain graphs

In general, given two disjoint sets of nodes  $S$  and  $C$ , a chain graph with components  $C$  and  $S$ , induced from a given DAG, is defined by one undirected or bidirected graph within the block of responses and by a directed graph between the components, in such a way that no partially directed cycles appear. This is the minimal chain graph compatible with the blocking and containing the set of distributions that are Markov to the generating DAG (see [Lauritzen and Richardson \(2002\)](#)). However, there are two different types of directed graphs from  $C$  to  $S$ . The first is the multivariate regression graph  $\mathcal{P}_{S|C}$  discussed in the previous section and the second is (in the terminology of [Wermuth and Cox \(2004\)](#)) the *blocked-concentration graph*,  $\mathcal{C}_{S|C}$ , which has an arrow  $i \rightarrow j$ , with  $i \in C$  and  $j \in S$ , iff the DAG does not imply  $i \perp\!\!\!\perp j \mid C \cup S \setminus \{i, j\}$ . Different types of chain graphs with different interpretations may be defined depending on the combinations of the chosen graphs for the regression and for the responses (see [Cox and Wermuth \(1996\)](#) and [Lauritzen and Richardson \(2002\)](#)):

- (a) edge matrices of the multivariate regression chain graph:  $\mathcal{S}_{SS|C}, \mathcal{P}_{S|C}$ ,
- (b) edge matrices of the chain graph with the Lauritzen-Wermuth-Frydenberg (LWF) Markov property:  $\mathcal{S}^{SS|C}, \mathcal{C}_{S|C}$ ,

- (c) edge matrices of the chain graph with the Andersson-Madigan-Perlman (AMP) Markov property:  $\mathcal{S}^{\mathcal{S}\mathcal{S}|C}, \mathcal{P}_{\mathcal{S}|C}$ .

The zeros in the adjacency matrix of the induced chain graph indicate the independencies that are logical consequences of the generating process underlying the directed acyclic graph. With the induced chain graphs we can easily assess the consequences implied by the DAG when some subsets of variables are considered jointly, with different conditioning set and also in different ordering.

*Example 9* Consider the DAG of Figure 8(a), with blocks of vertices  $C = \{W, U\}$  and  $S = \{X, Y, Z\}$ . Then, the induced chain graph with the LWF Markov interpretation is shown in Figure 8(b). The regression graph is the blocked concentration graph  $\mathcal{C}_{\mathcal{S}|C}$  and the undirected graph for  $S$  is the concentration graph of  $S$  given  $C$ . Thus, for example  $X \perp\!\!\!\perp Y|(Z, W, U)$  and  $X \perp\!\!\!\perp U|(W, Z, Y)$ . Instead, the induced chain graph with the alternative Markov property (AMP), shown in Figure 8(c), has an additional arrow  $W \rightarrow Z$  and the entailed independencies are, for instance,  $X \perp\!\!\!\perp Y|(W, U)$  and  $X \perp\!\!\!\perp U|W$ .

The R implementation of the above ideas is straightforward. We added to **ggm** a general function `inducedChainGraph` with arguments the adjacency matrix of the graph, a list `cc` of the chain components (with number of components  $> 1$ , and ordering from left to right) and a string `type` (with possible values "MRG" (multivariate regression graph), "AMP" and "LWF") for the required interpretation.

The required commands to obtain the chain graphs of Figure 8(b) and (c) are listed below.

```
> dag3 = DAG(X~W, W~Y, U~Y+Z)
> cc = list(c("W", "U"),
           c("X", "Y", "Z"))
> inducedChainGraph(dag3, cc=cc, type="LWF")
  W U X Y Z
W 0 1 1 1 0
U 1 0 0 1 1
X 0 0 0 0 0
Y 0 0 0 0 1
Z 0 0 0 1 0
> inducedChainGraph(dag3, cc=cc, type="AMP")
  W U X Y Z
W 0 1 1 1 1
U 1 0 0 1 1
X 0 0 0 0 0
Y 0 0 0 0 1
Z 0 0 0 1 0
```

The function `inducedChainGraph` can handle general situations, when the set  $V$  of vertices is partitioned into more than two chain components. For example, with three components  $a$ ,  $b$  and  $c$  (ordered from  $a$  to  $c$ ), the induced chain graphs, with multivariate regression, LWF and AMP interpretations have edge matrices, respectively:

$$\text{MRG : } \mathcal{S}_{bb|a}, \mathcal{P}_{b|a}, \mathcal{S}_{cc|ab}, \mathcal{P}_{c|ab} \quad (2)$$

```

> cc= list(c("U"),
           c("Z", "Y"),
           c("X", "W"))
> inducedChainGraph(dag3,cc,
                    type="MRG")
  U Z Y X W
U 0 1 1 0 0
Z 0 0 2 0 0
Y 0 2 0 1 1
X 0 0 0 0 2
W 0 0 0 2 0

```

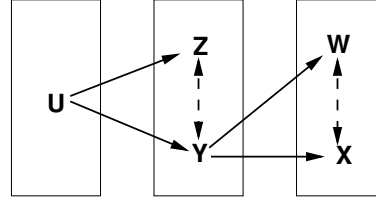


Figure 9: Chain graph of multivariate regression type induced by the DAG of Figure 8. The chain components are  $a = \{U\}$ ,  $b = \{Z, Y\}$  and  $c = \{W, X\}$ . The components graphs are listed in (2).

$$\text{LWF} : \mathcal{S}^{bb|a}, \mathcal{C}_{b|a}, \mathcal{S}^{cc|ab}, \mathcal{C}_{c|ab} \quad (3)$$

$$\text{AMP} : \mathcal{S}^{bb|a}, \mathcal{P}_{b|a}, \mathcal{S}^{cc|ab}, \mathcal{P}_{c|ab}. \quad (4)$$

*Example 10* In Figure 9 it is shown a chain graph of multivariate regression type with three components  $a = \{U\}$ ,  $b = \{Z, Y\}$  and  $c = \{W, X\}$  induced from the DAG of Figure 8.

## References

- Andersson S, Madigan D, Perlman M (2001). “Alternative Markov Properties for Chain Graphs.” *Scandinavian Journal of Statistics*, **28**, 33–85.
- Badsberg JH (2005). **dynamicGraph**: *An Interactive Graphical Tool for Manipulating Graphs*. R package version 0.2.0.1, URL <http://www.agrsci.org/>.
- Brito C, Pearl J (2002). “A New Identification Condition for Recursive Models With Correlated Errors.” *Structural Equation Modeling*, **9(4)**, 459–474.
- Cox D R, Wermuth N (1996). *Multivariate Dependencies. Models, Analysis and Interpretation*. Chapman and Hall, London.
- Frydenberg M (1990). “The Chain Markov Property.” *Scandinavian Journal of Statistics*, **17**, 333–353.
- Højsgaard S, Dethlefsen C (2005). “A Common Platform for Graphical Models in R: The **gRbase** Package.” *Journal of Statistical Software*, **14(17)**, 1–?
- Lauritzen S L (1996). *Graphical Models*. Oxford University Press, Oxford.
- Lauritzen S L (2002). “gRaphical Models in R.” *R News*, **3(2)**39.

- Lauritzen S L, Richardson T S (2002). “Chain Graph Models and their Causal Interpretations (with discussion).” *Journal of the Royal Statistical Society B*, **64**, 321–361.
- Marchetti GM, Drton M (2006). **ggm**: *Graphical Gaussian Models*. R package version 1.0.2.
- Pearl J (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA.
- Pearl J (2000). *Causality: Models, Reasoning and Inference*. Cambridge University Press, Cambridge.
- Richardson T, Spirtes P (2002). “Ancestral Graph Markov Models.” *Annals of Statistics*, **30**, 962–103.
- Verma T, Pearl J (1990). “Equivalence and Syntesis of Causal Models.” In M Henrion, L Shachter, L Kanal, J Lemmer (eds.), “Uncertainty in Artificial Intelligence: Proceedings of the Sixth Conference,” pp. 220–227. Morgan Kaufmann, San Francisco.
- Wermuth N, Cox D R (2004). “Joint Response Graphs and Separation Induced by Triangular Systems.” *Journal of the Royal Statistical Society B*, **66**, 687–717.

## A. Algorithms

All the induced graphs described in this paper can be obtained with explicit matrix expressions, see [Wermuth and Cox \(2004\)](#). Therefore the implementation in R is straightforward.

The formulae are based on the three matrix operators  $\text{In}[\cdot]$ ,  $\text{anc}[\cdot]$  and  $\text{clos}[\cdot]$ , defined for an edge matrix of a graph. In the following we will consider the edge matrix notation as equivalent to the graph.

- Given a square matrix  $M$ ,  $\text{In}[M]$  is the indicator matrix of the structural zeros in the matrix  $M$ . This may be computed in R with `abs(sign(M))`.
- The operator  $\text{anc}\mathcal{A}$  computes the transitive closure of a directed acyclic graph with edge matrix  $\mathcal{A}$ . The transitive closure of a DAG, defines a new directed acyclic graph by adding to the original one an edge  $i \rightarrow k$  whenever  $i \rightarrow j \rightarrow k$ .  
A simple close form of the transitive closure is  $\text{anc}\mathcal{A} = \text{In}[(2I - \mathcal{A})^{-1}]$ .
- The operator  $\text{clos}$  is defined for undirected graphs and is similar to  $\text{anc}$ . It adds to the original graph an edge  $i - k$  whenever  $i - j - k$ .

### A.1. Induced covariance graphs

The induced covariance graph for nodes in  $S$  given  $C$  has edge matrix

$$\mathcal{S}_{SS|C} = \text{In}[\text{anc}\mathcal{A}_{\bar{C}\bar{C}}\mathcal{W}(\text{anc}\mathcal{A}_{\bar{C}\bar{C}})^\top]_{SS} \quad (5)$$

where  $\bar{C} = V \setminus C$ ,  $\mathcal{T}_{ba} = \text{In}[\mathcal{A}_{ba}\text{anc}\mathcal{A}_{aa}]$  and  $\mathcal{W} = \text{clos}[\text{In}(I + \mathcal{T}_{CC}^\top\mathcal{T}_{CC})]$ .

We use matrix subscripts to define subsets of matrices. For example,  $\mathcal{A}_{aa}$  is the submatrix of  $\mathcal{A}$  obtained by selecting the rows and the columns indexed by  $a$ . We define analogously the matrix  $\mathcal{A}_{ab}$ .

Note that when  $a$  is empty the matrix  $\mathcal{A}_{ab}$  has zero rows. Moreover, we use the convention (standard in R and other matrix languages) that the products involving the matrix  $\mathcal{A}_{ab}$  return a zero matrix. For example, if  $a = \emptyset$  then we set  $\mathcal{A}_{ba}\mathcal{A}_{ab} = O_{bb}$ . With this convention, formula (5) is correct also when  $C$  is empty. Thus, the overall covariance graph is  $\text{In}[\text{anc}\mathcal{A}(\text{anc}\mathcal{A})^\top]$  while the covariance graph for  $S$  marginalizing over  $V \setminus S$  is

$$\mathcal{S}_{SS} = \text{In}[\text{anc}\mathcal{A}(\text{anc}\mathcal{A})^\top]_{SS}$$

because the middle factor in (5) turns out to be in this case an identity matrix.

The induced covariance graph can be used to verify if two sets  $A$  and  $B$  are d-separated given  $C$  in the DAG. For the global Markov property this happens iff  $A \perp\!\!\!\perp B | C$  in the DAG that is iff the induced covariance graph  $\mathcal{S}_{SS|C}$  of  $S = A \cup B$  given  $C$  is such that the submatrix  $[\mathcal{S}_{SS|C}]_{A,B} = O$ . Actually, the function `dSep` follows this approach to test d-separation.

### A.2. Induced concentration graph

The edge matrix of the induced concentration graph  $\mathcal{S}^{SS|C}$  for  $S$  given  $C$

$$\mathcal{S}^{SS|C} = \text{In}[\mathcal{A}_{\bar{M}\bar{M}.M}^\top \mathcal{Q}\mathcal{A}_{\bar{M}\bar{M}.M}]_{S,S} \quad (6)$$

where  $M = V \setminus (S \cup C)$ , are the nodes marginalized over,  $\bar{M} = V \setminus M$ ,  $\mathcal{A}_{bb.a} = \text{In}[\mathcal{A}_{bb} + \mathcal{A}_{ba}\text{anc}\mathcal{A}_{aa}\mathcal{A}_{ab}]$  and  $\mathcal{Q} = \text{clos}[\text{In}(I + \mathcal{T}_{\bar{M}\bar{M}}\mathcal{T}_{\bar{M}\bar{M}}^\top)]$ .

### A.3. Induced regression graphs

The edge matrix of the induced multivariate regression graph of  $S$  given  $C$  is

$$\mathcal{P}_{S|C} = \text{In}[\text{anc}\mathcal{A}_{\bar{C}\bar{C}}\mathcal{T}_{\bar{C}\bar{C}}^\top\mathcal{D}\mathcal{A}_{CC.\bar{C}} + \mathcal{F}_{\bar{C}\bar{C}}]_{SC} \quad (7)$$

where  $\mathcal{D} = \text{clos}[\text{In}(I + \mathcal{T}_{\bar{C}\bar{C}}\mathcal{T}_{\bar{C}\bar{C}}^\top)]$ ,  $\mathcal{F}_{ab} = \text{In}[\text{anc}\mathcal{A}_{aa}\mathcal{A}_{ab}]$ .

Finally, the edge matrix of the induced blocked-concentration graph is obtained from the induced concentration graph for the nodes  $C \cup S$ , with empty conditioning set, and then extracting the  $(S, C)$  block:

$$\mathcal{C}_{S|C} = [\mathcal{S}^{(CUS)(CUS)}]_{SC}. \quad (8)$$

#### Affiliation:

Giovanni M. Marchetti  
 Dipartimento di Statistica ‘‘G. Parenti’’  
 University of Florence  
 50134 Firenze, Italy  
 E-mail: [giovanni.marchetti@unifi.it](mailto:giovanni.marchetti@unifi.it)  
 URL: <http://www.ds.unifi.it/gmm/>