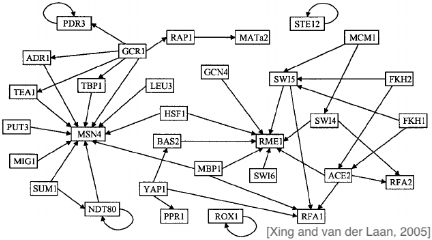
Learning Linear Non-Gaussian Causal Models via Algebraic Constraints

Elina Robeva
The University of British Columbia

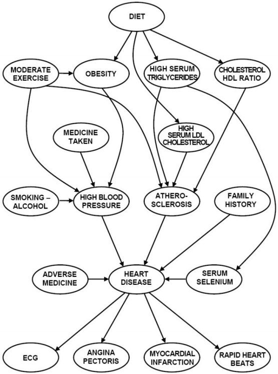
April 21, 2023

Causal models



[Xing and van der Laan, 2005]

Gene regulatory networks



[Barbini, Manzi, Barbini 2013]

Disease diagnosis graphs

How can we learn the structure of these graphs from observations?

Structural causal models

Definition

A *structural causal model* consists of a directed acyclic graph (DAG) $G = (V, E)$, and a set of equations/assignments between the random variables $\{X_v : v \in V\}$:

$$X_v = f_v(X_{\mathrm{pa}(v)}, \varepsilon_v), \;\; v \in V$$

where $X_{\mathrm{pa}(v)} = (X_u : u \to v \in E)$ and ε_v is noise such that $\{\varepsilon_v : v \in V\}$ are independent noise terms.

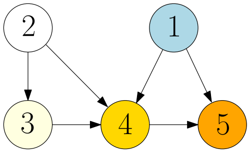
Structural causal models

Definition

A *structural causal model* consists of a directed acyclic graph (DAG) $G = (V, E)$, and a set of equations/assignments between the random variables $\{X_v : v \in V\}$:

$$X_v = f_v(X_{\mathrm{pa}(v)}, \varepsilon_v),\ \ v \in V$$

where $X_{\mathrm{pa}(v)} = (X_u : u \to v \in E)$ and ε_v is noise such that $\{\varepsilon_v : v \in V\}$ are independent noise terms.



$$\begin{aligned}
X_1 &= f_1(\varepsilon_1) \\
X_2 &= f_2(\varepsilon_2) \\
X_3 &= f_3(X_2, \varepsilon_3) \\
X_4 &= f_4(X_1, X_2, X_3, \varepsilon_4) \\
X_5 &= f_5(X_1, X_4, \varepsilon_5).
\end{aligned}$$

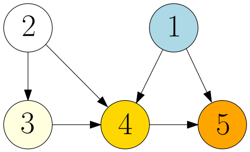
Structural causal models

Definition

A *structural causal model* consists of a directed acyclic graph (DAG) $G = (V, E)$, and a set of equations/assignments between the random variables $\{X_v : v \in V\}$:

$$X_v = f_v(X_{\mathrm{pa}(v)}, \varepsilon_v), \;\; v \in V$$

where $X_{\mathrm{pa}(v)} = (X_u : u \to v \in E)$ and ε_v is noise such that $\{\varepsilon_v : v \in V\}$ are independent noise terms.



$$\begin{aligned} X_1 &= f_1(\varepsilon_1) \\ X_2 &= f_2(\varepsilon_2) \\ X_3 &= f_3(X_2, \varepsilon_3) \\ X_4 &= f_4(X_1, X_2, X_3, \varepsilon_4) \\ X_5 &= f_5(X_1, X_4, \varepsilon_5). \end{aligned}$$

Given samples $X^{(1)}, \ldots, X^{(n)} \in \mathbb{R}^{|V|}$ arising from such a model, can we identify G?

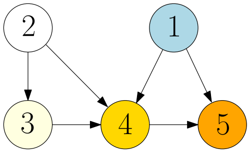
Structural causal models

Definition

A *structural causal model* consists of a directed acyclic graph (DAG) $G = (V, E)$, and a set of equations/assignments between the random variables $\{X_v : v \in V\}$:

$$X_v = f_v(X_{\mathrm{pa}(v)}, \varepsilon_v), \ \ v \in V$$

where $X_{\mathrm{pa}(v)} = (X_u : u \to v \in E)$ and ε_v is noise such that $\{\varepsilon_v : v \in V\}$ are independent noise terms.

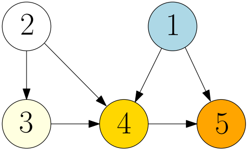


$$\begin{aligned} X_1 &= f_1(\varepsilon_1) \\ X_2 &= f_2(\varepsilon_2) \\ X_3 &= f_3(X_2, \varepsilon_3) \\ X_4 &= f_4(X_1, X_2, X_3, \varepsilon_4) \\ X_5 &= f_5(X_1, X_4, \varepsilon_5). \end{aligned}$$

Given samples $X^{(1)}, \ldots, X^{(n)} \in \mathbb{R}^{|V|}$ arising from such a model, can we identify G?

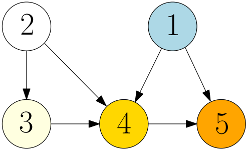
- Linear structural equation models

Linear structural equation models



$$
\begin{aligned}
X_1 &= \varepsilon_1 \\
X_2 &= \varepsilon_2 \\
X_3 &= \lambda_{23} X_2 + \varepsilon_3 \\
X_4 &= \lambda_{14} X_1 + \lambda_{24} X_2 + \lambda_{34} X_3 + \varepsilon_4 \\
X_5 &= \lambda_{15} X_1 + \lambda_{45} X_4 + \varepsilon_5.
\end{aligned}
$$

Linear structural equation models

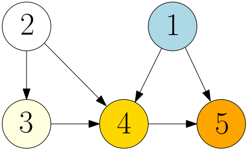


$$X_1 = \varepsilon_1$$
$$X_2 = \varepsilon_2$$
$$X_3 = \lambda_{23} X_2 + \varepsilon_3$$
$$X_4 = \lambda_{14} X_1 + \lambda_{24} X_2 + \lambda_{34} X_3 + \varepsilon_4$$
$$X_5 = \lambda_{15} X_1 + \lambda_{45} X_4 + \varepsilon_5.$$

For a general directed acyclic graph $G = (V, E)$, the *linear structural equation model corresponding to* G consists of the the graph G and the linear equations

$$X_i = \sum_{j \in \mathrm{pa}(i)} \lambda_{ji} X_j + \varepsilon_i, \qquad \text{where the variables } \{\varepsilon_i\}_{i \in V} \text{ are independent.}$$

Linear structural equation models



$$
\begin{aligned}
X_1 &= \varepsilon_1 \\
X_2 &= \varepsilon_2 \\
X_3 &= \lambda_{23} X_2 + \varepsilon_3 \\
X_4 &= \lambda_{14} X_1 + \lambda_{24} X_2 + \lambda_{34} X_3 + \varepsilon_4 \\
X_5 &= \lambda_{15} X_1 + \lambda_{45} X_4 + \varepsilon_5.
\end{aligned}
$$

For a general directed acyclic graph $G = (V, E)$, the *linear structural equation model corresponding to* G consists of the the graph G and the linear equations

$$X_i = \sum_{j \in \mathrm{pa}(i)} \lambda_{ji} X_j + \varepsilon_i, \qquad \text{where the variables } \{\varepsilon_i\}_{i \in V} \text{ are independent.}$$

In matrix-vector form

$$X = \Lambda^T X + \varepsilon.$$

Equivalently,

$$X = (I - \Lambda)^{-T} \varepsilon.$$

Linear Gaussian models

$$X_i = \sum_{j \in \mathrm{pa}(i)} \lambda_{ji} X_j + \epsilon_i, \quad \text{where } \epsilon \sim \mathcal{N}(\nu, \Omega), \text{ and } \Omega = \mathrm{diag}(\omega_1, \ldots, \omega_n),$$

Linear Gaussian models

$$X_i = \sum_{j \in \mathrm{pa}(i)} \lambda_{ji} X_j + \epsilon_i, \qquad \text{where } \epsilon \sim \mathcal{N}(\nu, \Omega), \text{ and } \Omega = \mathrm{diag}(\omega_1, \ldots, \omega_n),$$

$$X = (I - \Lambda)^{-T} \epsilon.$$

Linear Gaussian models

$$X_i = \sum_{j \in \mathrm{pa}(i)} \lambda_{ji} X_j + \epsilon_i, \qquad \text{where } \epsilon \sim \mathcal{N}(\nu, \Omega), \text{ and } \Omega = \mathrm{diag}(\omega_1, \ldots, \omega_n),$$

$$X = (I - \Lambda)^{-T} \epsilon.$$

Thus, $X \sim \mathcal{N}(\mu, \Sigma)$, where

$$\Sigma = (I - \Lambda)^{-T} \Omega (I - \Lambda)^{-1}.$$

The set of distributions $\mathcal{M}_G$ arising from a Gaussian linear causal model with DAG $G = (V, E)$ is called the **directed Gaussian graphical model corresponding to G**, and

$$\mathcal{M}_G = \{\Sigma \;:\; \Sigma = (I - \Lambda)^{-T} \Omega (I - \Lambda)^{-1}, \; \Lambda \in \mathbb{R}^E, \Omega \succ 0 \text{ diagonal}\}.$$

Linear Gaussian models

$$X_i = \sum_{j \in \mathrm{pa}(i)} \lambda_{ji} X_j + \epsilon_i, \quad \text{where } \epsilon \sim \mathcal{N}(\nu, \Omega), \text{ and } \Omega = \mathrm{diag}(\omega_1, \ldots, \omega_n),$$

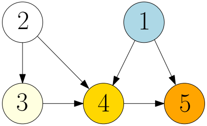
$$X = (I - \Lambda)^{-T} \epsilon.$$

Thus, $X \sim \mathcal{N}(\mu, \Sigma)$, where

$$\Sigma = (I - \Lambda)^{-T} \Omega (I - \Lambda)^{-1}.$$

The set of distributions $\mathcal{M}_G$ arising from a Gaussian linear causal model with DAG $G = (V, E)$ is called the **directed Gaussian graphical model corresponding to G**, and

$$\mathcal{M}_G = \{\Sigma \ : \ \Sigma = (I - \Lambda)^{-T} \Omega (I - \Lambda)^{-1}, \ \Lambda \in \mathbb{R}^E, \Omega \succ 0 \text{ diagonal}\}.$$



$$\Sigma = (I - \Lambda)^{-T} \Omega (I - \Lambda)^{-1}, \text{ where}$$

$$\Lambda = \begin{pmatrix} 0 & 0 & 0 & \lambda_{14} & \lambda_{15} \\ 0 & 0 & \lambda_{23} & \lambda_{24} & 0 \\ 0 & 0 & 0 & \lambda_{34} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{45} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \Omega = \begin{pmatrix} \omega_{11} & 0 & 0 & 0 & 0 \\ 0 & \omega_{22} & 0 & 0 & 0 \\ 0 & 0 & \omega_{33} & 0 & 0 \\ 0 & 0 & 0 & \omega_{44} & 0 \\ 0 & 0 & 0 & 0 & \omega_{55} \end{pmatrix}.$$

Linear Gaussian models

$$X_i = \sum_{j \in \mathrm{pa}(i)} \lambda_{ji} X_j + \epsilon_i, \quad \text{where } \epsilon \sim \mathcal{N}(\nu, \Omega), \text{ and } \Omega = \mathrm{diag}(\omega_1, \dots, \omega_n),$$

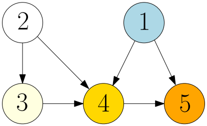
$$X = (I - \Lambda)^{-T} \epsilon.$$

Thus, $X \sim \mathcal{N}(\mu, \Sigma)$, where

$$\Sigma = (I - \Lambda)^{-T} \Omega (I - \Lambda)^{-1}.$$

The set of distributions $\mathcal{M}_G$ arising from a Gaussian linear causal model with DAG $G = (V, E)$ is called the **directed Gaussian graphical model corresponding to G**, and

$$\mathcal{M}_G = \{\Sigma \ : \ \Sigma = (I - \Lambda)^{-T} \Omega (I - \Lambda)^{-1}, \ \Lambda \in \mathbb{R}^E, \Omega \succ 0 \text{ diagonal}\}.$$



$$\Sigma = (I - \Lambda)^{-T} \Omega (I - \Lambda)^{-1}, \text{ where}$$

$$\Lambda = \begin{pmatrix} 0 & 0 & 0 & \lambda_{14} & \lambda_{15} \\ 0 & 0 & \lambda_{23} & \lambda_{24} & 0 \\ 0 & 0 & 0 & \lambda_{34} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{45} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \Omega = \begin{pmatrix} \omega_{11} & 0 & 0 & 0 & 0 \\ 0 & \omega_{22} & 0 & 0 & 0 \\ 0 & 0 & \omega_{33} & 0 & 0 \\ 0 & 0 & 0 & \omega_{44} & 0 \\ 0 & 0 & 0 & 0 & \omega_{55} \end{pmatrix}.$$

- Markov equivalence: $\mathcal{M}_G = \mathcal{M}_H \implies$ cannot identify the graph G uniquely.

Non-Gaussian Linear Structural Equation Models

Given a DAG $G = (V, E)$,

$$X = (I - \Lambda)^{-T}\varepsilon, \qquad \text{where the } \{\varepsilon_v\}_{v \in V} \text{ are independent and } \Lambda \in \mathbb{R}^E.$$

Non-Gaussian Linear Structural Equation Models

Given a DAG $G = (V, E)$,

$$X = (I - \Lambda)^{-T}\varepsilon, \quad \text{where the } \{\varepsilon_v\}_{v \in V} \text{ are independent and } \Lambda \in \mathbb{R}^E.$$

- **Independent component analysis:** Given $X = A\varepsilon$, where ε is a vector of independent components, want to recover A,
  ... up to permutation and scaling of its columns.

Non-Gaussian Linear Structural Equation Models

Given a DAG $G = (V, E)$,

$$X = (I - \Lambda)^{-T}\varepsilon, \qquad \text{where the } \{\varepsilon_v\}_{v \in V} \text{ are independent and } \Lambda \in \mathbb{R}^E.$$

- **Independent component analysis:** Given $X = A\varepsilon$, where ε is a vector of independent components, want to recover A,
  ... up to permutation and scaling of its columns.
- If at least two ε_j are Gaussian, then recovering A uniquely is impossible.

Non-Gaussian Linear Structural Equation Models

Given a DAG $G = (V, E)$,

$$X = (I - \Lambda)^{-T}\varepsilon, \qquad \text{where the } \{\varepsilon_v\}_{v \in V} \text{ are independent and } \Lambda \in \mathbb{R}^E.$$

- **Independent component analysis:** Given $X = A\varepsilon$, where ε is a vector of independent components, want to recover A,
  ... up to permutation and scaling of its columns.
- If at least two ε_j are Gaussian, then recovering A uniquely is impossible.

Theorem (Comon and Jutten, *Handbook of Blind Source Separation*, 2010)

If all (or all but one) ε_j are non-Gaussian, A can be recovered (up to permutation and scaling).

Non-Gaussian Linear Structural Equation Models

Given a DAG $G = (V, E)$,

$$X = (I - \Lambda)^{-T}\varepsilon, \qquad \text{where the } \{\varepsilon_v\}_{v \in V} \text{ are independent and } \Lambda \in \mathbb{R}^E.$$

- **Independent component analysis:** Given $X = A\varepsilon$, where ε is a vector of independent components, want to recover A,
  ... up to permutation and scaling of its columns.
- If at least two ε_j are Gaussian, then recovering A uniquely is impossible.

Theorem (Comon and Jutten, *Handbook of Blind Source Separation*, 2010)

If all (or all but one) ε_j are non-Gaussian, A can be recovered (up to permutation and scaling).

- ICA Methods: maximum likelihood estimation, 4th order cumulant tensor decomposition, maximizing |kurtosis| of $A^{-1}X$ (a measure of non-Gaussianity)

Linear Non-Gaussian Acyclic Models (LiNGAM)

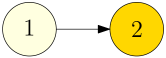
$$X = (I - \Lambda)^{-T} \varepsilon.$$

- Shimizu et al., 2006: LiNGAM; use ICA methods; estimate of $(I - \Lambda)$ has all entries non-zero
- Shimizu et al., 2011: Direct-LiNGAM; a source node is independent from regression residuals; does not work if #observations $<$ #variables (high-dimensions)
- Wang and Drton, 2018: High-dimensional algorithm, exploits *relationships between second and higher order moments of X*

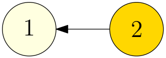
Linear Non-Gaussian Acyclic Models (LiNGAM)

$$X = (I - \Lambda)^{-T}\varepsilon.$$

- Shimizu et al., 2006: LiNGAM; use ICA methods; estimate of $(I - \Lambda)$ has all entries non-zero
- Shimizu et al., 2011: Direct-LiNGAM; a source node is independent from regression residuals; does not work if #observations $<$ #variables (high-dimensions)
- Wang and Drton, 2018: High-dimensional algorithm, exploits *relationships between second and higher order moments of X*



$$\mathbb{E}[X_1X_2]\mathbb{E}[X_1^3] - \mathbb{E}[X_1^2]\mathbb{E}[X_1^2X_2] = 0.$$



$$\mathbb{E}[X_1X_2]\mathbb{E}[X_1^3] - \mathbb{E}[X_1^2]\mathbb{E}[X_1^2X_2] \neq 0$$

generically, in particular, third order moments need to be non-Gaussian.

Looking at higher moments

$$X = (I - \Lambda)^{-T}\varepsilon.$$

Definition

The *linear structural equation model* $\mathcal{M}^{(2,3)}(G)$ of second and third order moments corresponding to a DAG $G = (V, E)$ with $|V| = n$ is defined as

$$\begin{aligned}\mathcal{M}^{(2,3)}(G) = \{(S = (I - \Lambda)^{-T}\Omega^{(2)}(I - \Lambda)^{-1},\\ T = \Omega^{(3)} \bullet (I - \Lambda)^{-1} \bullet (I - \Lambda)^{-1} \bullet (I - \Lambda)^{-1}) :\\ \Omega^{(2)} \text{ is } n \times n \text{ positive definite diagonal matrix},\\ \Omega^{(3)} \text{ is } n \times n \times n \text{ diagonal 3-way tensor, and } \Lambda \in \mathbb{R}^E\}.\end{aligned}$$

Here, $\bullet$ denotes the *Tucker product*.

Theorem (Améndola, Drton, Grosdos, Homs-Pons, and R., 2021+)

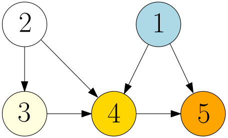
The set of second and third order moments (T, S) *of a linear non-Gaussian causal model corresponding to a* <u>*tree*</u> *DAG are precisely the ones that satisfy certain quadratic binomials which arise as the* 2×2 *minors of certain matrices constructed from the DAG.*

- $s_{ij} = 0$ for all $i, j \in V$ for which there is no 2-trek between i and j;
- $t_{ijk} = 0$ for all $i, j, k \in V$ for which there is no 2-trek between i, j, k;
- the 2×2 minors of the matrix A_{ij} are 0 whenever there is a path from i to j, where

$$A_{ij} = \begin{bmatrix} s_{ik_1} & \cdots & s_{ik_r} & t_{i\ell_1 m_1} & \cdots & t_{i\ell_q m_q} \\ s_{jk_1} & \cdots & s_{jk_r} & t_{j\ell_1 m_1} & \cdots & t_{j\ell_q m_q} \end{bmatrix},$$

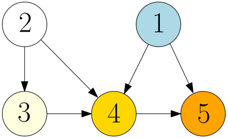
  where
  - $k_1, \ldots, k_r$ are all vertices such that $\text{top}(i, k_a) = \text{top}(j, k_a)$ and
  - (l_1, m_1),...,(l_q, m_q) are all pairs of vertices such that $\text{top}(i, l_b, m_b) = \text{top}(j, l_b, m_b)$.

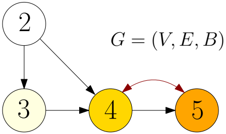
Introducing hidden variables



$$X_1 = \epsilon_1$$
$$X_2 = \epsilon_2$$
$$X_3 = \lambda_{23} X_2 + \epsilon_3$$
$$X_4 = \lambda_{14} X_1 + \lambda_{24} X_2 + \lambda_{34} X_3 + \epsilon_4$$
$$X_5 = \lambda_{15} X_1 + \lambda_{45} X_4 + \epsilon_5$$

Introducing hidden variables





$X_1 = \epsilon_1$

$X_2 = \epsilon_2$

$X_3 = \lambda_{23} X_2 + \epsilon_3$

$X_4 = \lambda_{14} X_1 + \lambda_{24} X_2 + \lambda_{34} X_3 + \epsilon_4$

$X_5 = \lambda_{15} X_1 + \lambda_{45} X_4 + \epsilon_5$

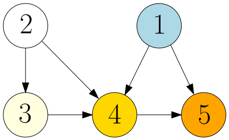
$X_2 = \epsilon_2$

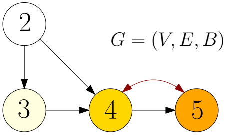
$X_3 = \lambda_{23} X_2 + \epsilon_3$

$X_4 = \lambda_{24} X_2 + \lambda_{34} X_3 + \tilde{\epsilon}_4$

$X_5 = \lambda_{45} X_4 + \tilde{\epsilon}_5$

Introducing hidden variables





$$\begin{aligned} X_1 &= \epsilon_1 \\ X_2 &= \epsilon_2 \\ X_3 &= \lambda_{23} X_2 + \epsilon_3 \\ X_4 &= \lambda_{14} X_1 + \lambda_{24} X_2 + \lambda_{34} X_3 + \epsilon_4 \\ X_5 &= \lambda_{15} X_1 + \lambda_{45} X_4 + \epsilon_5 \end{aligned}$$

$$\Sigma = (I - \Lambda)^{-T} \Omega (I - \Lambda)^{-1},$$

$$\text{where } \Lambda = \begin{pmatrix} 0 & 0 & 0 & \lambda_{14} & \lambda_{15} \\ 0 & 0 & \lambda_{23} & \lambda_{24} & 0 \\ 0 & 0 & 0 & \lambda_{34} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{45} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^E,$$

$$\Omega = \begin{pmatrix} \omega_{11} & 0 & 0 & 0 & 0 \\ 0 & \omega_{22} & 0 & 0 & 0 \\ 0 & 0 & \omega_{33} & 0 & 0 \\ 0 & 0 & 0 & \omega_{44} & 0 \\ 0 & 0 & 0 & 0 & \omega_{55} \end{pmatrix} \in \text{PD}.$$

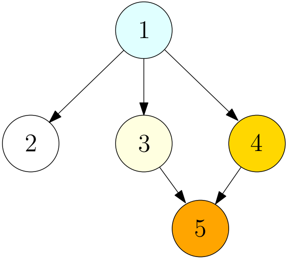
$$\begin{aligned} X_2 &= \epsilon_2 \\ X_3 &= \lambda_{23} X_2 + \epsilon_3 \\ X_4 &= \lambda_{24} X_2 + \lambda_{34} X_3 + \tilde{\epsilon}_4 \\ X_5 &= \lambda_{45} X_4 + \tilde{\epsilon}_5 \end{aligned}$$

$$\Sigma = (I - \Lambda)^{-T} \Omega (I - \Lambda)^{-1},$$

$$\text{where } \Lambda = \begin{pmatrix} 0 & \lambda_{23} & \lambda_{24} & 0 \\ 0 & 0 & \lambda_{34} & 0 \\ 0 & 0 & 0 & \lambda_{45} \\ 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^E,$$

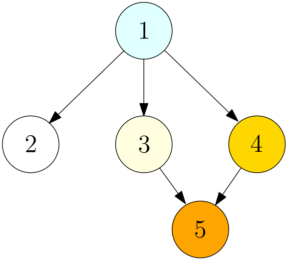
$$\Omega = \begin{pmatrix} \omega_{22} & 0 & 0 & 0 \\ 0 & \omega_{33} & 0 & 0 \\ 0 & 0 & \omega_{44} & \omega_{45} \\ 0 & 0 & \omega_{45} & \omega_{55} \end{pmatrix} \in \text{PD}^B.$$

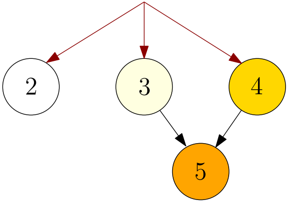
Introducing hidden variables



$$X_1 = \varepsilon_1$$
$$X_2 = \lambda_{12}X_1 + \varepsilon_2$$
$$X_3 = \lambda_{13}X_1 + \varepsilon_3$$
$$X_4 = \lambda_{14}X_1 + \varepsilon_4$$
$$X_5 = \lambda_{35}X_3 + \lambda_{45}X_4 + \varepsilon_5.$$

Introducing hidden variables



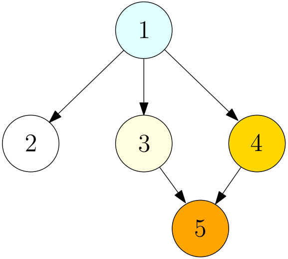


$$X_1 = \varepsilon_1$$
$$X_2 = \lambda_{12} X_1 + \varepsilon_2$$
$$X_3 = \lambda_{13} X_1 + \varepsilon_3$$
$$X_4 = \lambda_{14} X_1 + \varepsilon_4$$
$$X_5 = \lambda_{35} X_3 + \lambda_{45} X_4 + \varepsilon_5.$$

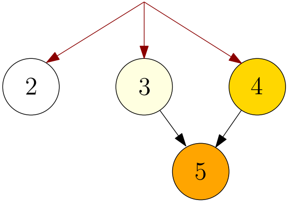
$$X_2 = \tilde{\varepsilon}_2$$
$$X_3 = \tilde{\varepsilon}_3$$
$$X_4 = \tilde{\varepsilon}_4$$
$$X_5 = \lambda_{35} X_3 + \lambda_{45} X_4 + \varepsilon_5,$$

where $\varepsilon_5 \perp\!\!\!\perp \tilde{\varepsilon}_2, \tilde{\varepsilon}_3, \tilde{\varepsilon}_4$.

Introducing hidden variables



$$X_1 = \varepsilon_1$$
$$X_2 = \lambda_{12} X_1 + \varepsilon_2$$
$$X_3 = \lambda_{13} X_1 + \varepsilon_3$$
$$X_4 = \lambda_{14} X_1 + \varepsilon_4$$
$$X_5 = \lambda_{35} X_3 + \lambda_{45} X_4 + \varepsilon_5.$$



$$X_2 = \tilde{\varepsilon}_2$$
$$X_3 = \tilde{\varepsilon}_3$$
$$X_4 = \tilde{\varepsilon}_4$$
$$X_5 = \lambda_{35} X_3 + \lambda_{45} X_4 + \varepsilon_5,$$

where $\varepsilon_5 \perp\!\!\!\perp \tilde{\varepsilon}_2, \tilde{\varepsilon}_3, \tilde{\varepsilon}_4$.

The new graph $G = (V, E, H)$ has directed edges E and *multi-directed edges* H.

Learning LiNGAMs with hidden variables from observational data

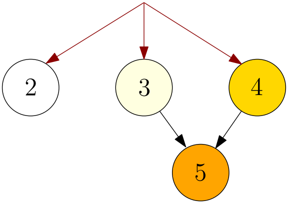
Existing methods for learning $G = (V, E, H)$ either

- Use ICA methods (Hoyer et al., 2008), which don't guarantee convergence to a global optimum, OR
- Only learn a graph $G = (V, E, B)$ with directed and *bidirected* edges (ParcelLiNGAM, Tashiro et al., 2014, Wang and Drton, 2020).

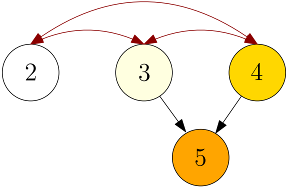
Learning LiNGAMs with hidden variables from observational data

Existing methods for learning $G = (V, E, H)$ either

- Use ICA methods (Hoyer et al., 2008), which don't guarantee convergence to a global optimum, OR
- Only learn a graph $G = (V, E, B)$ with directed and *bidirected* edges (ParcelLiNGAM, Tashiro et al., 2014, Wang and Drton, 2020).



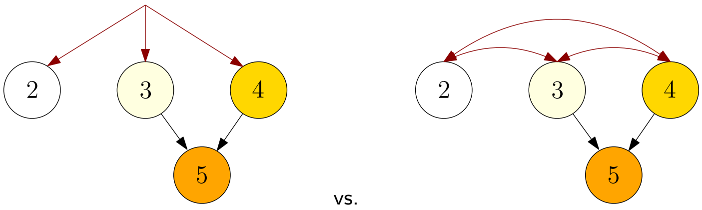
vs.



Learning LiNGAMs with hidden variables from observational data

Existing methods for learning $G = (V, E, H)$ either

- Use ICA methods (Hoyer et al., 2008), which don't guarantee convergence to a global optimum, OR
- Only learn a graph $G = (V, E, B)$ with directed and *bidirected* edges (ParcelLiNGAM, Tashiro et al., 2014, Wang and Drton, 2020).



- (Liu, Robeva, Wang, 2020): Learn $G = (V, E, H)$, where H has multidirected edges; G is a bow-free acyclic graph; use high-order cumulant information

Vanishing of cumulants

- For a zero-mean random vector $X = (X_1, \dots, X_d)$, its *k-th order cumulant* is an $d \times \cdots \times d$ (k times) tensor $C^{(k)}$ whose entries can be obtained from the moments of X, e.g. for $k = 4$:

$$C^{(4)}_{i_1,i_2,i_3,i_4} = \mathbb{E}[X_{i_1}X_{i_2}X_{i_3}X_{i_4}] - \mathbb{E}[X_{i_1}X_{i_2}]\mathbb{E}[X_{i_3}X_{i_4}] - \mathbb{E}[X_{i_1}X_{i_3}]\mathbb{E}[X_{i_2}X_{i_4}] - \mathbb{E}[X_{i_1}X_{i_4}]\mathbb{E}[X_{i_2}X_{i_3}].$$

Vanishing of cumulants

- For a zero-mean random vector $X = (X_1, \ldots, X_d)$, its *k-th order cumulant* is an $d \times \cdots \times d$ (k times) tensor $C^{(k)}$ whose entries can be obtained from the moments of X, e.g. for $k = 4$:

$$C^{(4)}_{i_1,i_2,i_3,i_4} = \mathbb{E}[X_{i_1}X_{i_2}X_{i_3}X_{i_4}] - \mathbb{E}[X_{i_1}X_{i_2}]\mathbb{E}[X_{i_3}X_{i_4}] - \mathbb{E}[X_{i_1}X_{i_3}]\mathbb{E}[X_{i_2}X_{i_4}] - \mathbb{E}[X_{i_1}X_{i_4}]\mathbb{E}[X_{i_2}X_{i_3}].$$

Theorem (Robeva and Seby, 2020)

If X comes from a linear non-Gaussian acyclic model with graph $G = (V, E, H)$ and X has cumulants $C^{(k)}$, then

$$C^{(k)}_{i_1,\ldots,i_k} = 0$$

if and only if there is no k-trek between the vertices *$i_1, \ldots, i_k$ in G.*

Vanishing of cumulants

- For a zero-mean random vector $X = (X_1, \dots, X_d)$, its *k-th order cumulant* is an $d \times \cdots \times d$ (k times) tensor $C^{(k)}$ whose entries can be obtained from the moments of X, e.g. for $k = 4$:

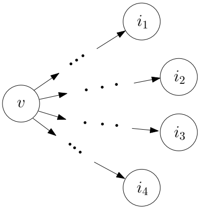
$$C^{(4)}_{i_1,i_2,i_3,i_4} = \mathbb{E}[X_{i_1}X_{i_2}X_{i_3}X_{i_4}] - \mathbb{E}[X_{i_1}X_{i_2}]\mathbb{E}[X_{i_3}X_{i_4}] - \mathbb{E}[X_{i_1}X_{i_3}]\mathbb{E}[X_{i_2}X_{i_4}] - \mathbb{E}[X_{i_1}X_{i_4}]\mathbb{E}[X_{i_2}X_{i_3}].$$

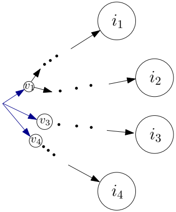
Theorem (Robeva and Seby, 2020)

If X comes from a linear non-Gaussian acyclic model with graph $G = (V, E, H)$ and X has cumulants $C^{(k)}$, then

$$C^{(k)}_{i_1,\dots,i_k} = 0$$

if and only if there is no k-trek between the vertices *$i_1, \dots, i_k$ in G.*





k-treks

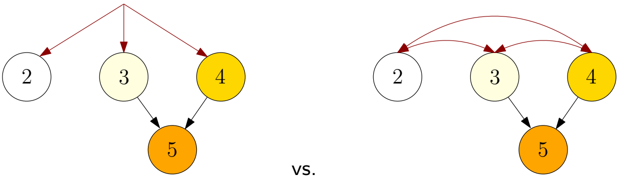
Theorem (Robeva and Seby, 2020)

If X comes from a linear non-Gaussian causal model with graph $G = (V, E, H)$ and X has cumulants $C^{(k)}$, then

$$C^{(k)}_{i_1,\ldots,i_k} = 0$$

if and only if there is no k-trek between the vertices *$i_1, \ldots, i_k$ in G.*

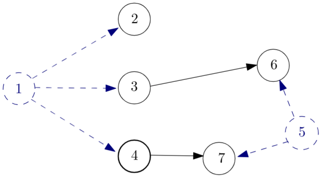
- Thus, we can distinguish:



$C^{(3)}_{2,3,4} \neq 0$ vs. $C^{(3)}_{2,3,4} = 0.$

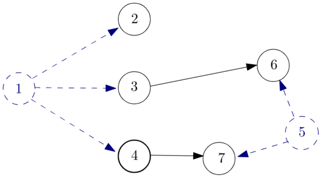
- Get and algorithm to learn $G = (V, E, H)$ based on high-order cumulants.

Learning $G = (V, E, H)$ [Liu, Robeva, Wang, 2020]

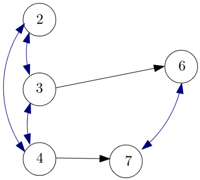


1. Obtain samples $Y = (Y^{(1)}, \ldots, Y^{(N)})$ from LiNGAM with unknown $G = (V, E, H)$

Learning $G = (V, E, H)$ [Liu, Robeva, Wang, 2020]

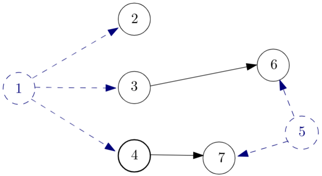


1. Obtain samples $Y = (Y^{(1)}, \ldots, Y^{(N)})$ from LiNGAM with unknown $G = (V, E, H)$

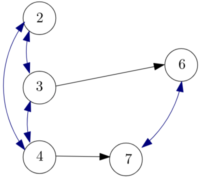


2. Learn a LiNGAM with graph (V, E, B) with bidirected edges and coefficient matrix Λ, e.g. using [Wang and Drton, 2020]

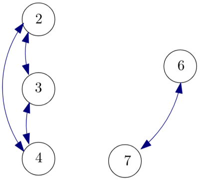
Learning $G = (V, E, H)$ [Liu, Robeva, Wang, 2020]



1. Obtain samples $Y = (Y^{(1)}, \ldots, Y^{(N)})$ from LiNGAM with unknown $G = (V, E, H)$

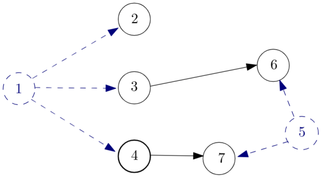


2. Learn a LiNGAM with graph (V, E, B) with bidirected edges and coefficient matrix Λ, e.g. using [Wang and Drton, 2020]

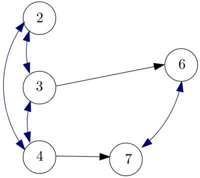


3. "Remove" directed edges E via $X = Y - \Lambda^T Y$.

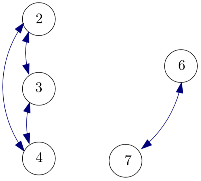
Learning $G = (V, E, H)$ [Liu, Robeva, Wang, 2020]



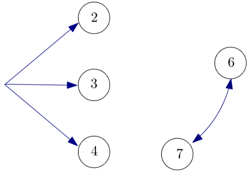
1. Obtain samples $Y = (Y^{(1)}, \ldots, Y^{(N)})$ from LiNGAM with unknown $G = (V, E, H)$



2. Learn a LiNGAM with graph (V, E, B) with bidirected edges and coefficient matrix Λ, e.g. using [Wang and Drton, 2020]

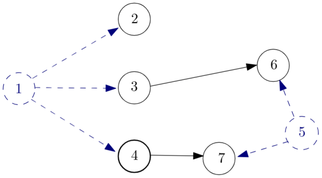


3. "Remove" directed edges E via $X = Y - \Lambda^T Y$.

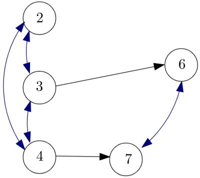


4. Identify the multidirected edges H by "merging" some of the bidirected edges in graph $(V, \emptyset, B)$ using the cumulants of X.

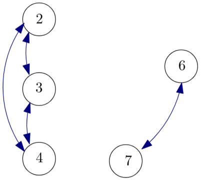
Learning $G = (V, E, H)$ [Liu, Robeva, Wang, 2020]



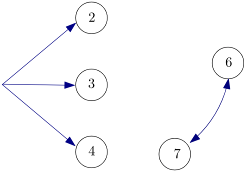
1. Obtain samples $Y = (Y^{(1)}, \dots, Y^{(N)})$ from LiNGAM with unknown $G = (V, E, H)$



2. Learn a LiNGAM with graph (V, E, B) with bidirected edges and coefficient matrix Λ, e.g. using [Wang and Drton, 2020]



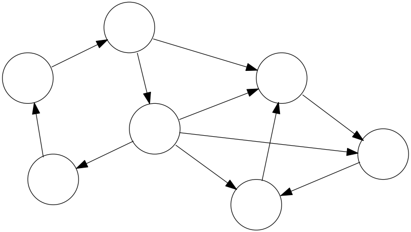
3. "Remove" directed edges E via $X = Y - \Lambda^T Y$.



4. Identify the multidirected edges H by "merging" some of the bidirected edges in graph $(V, \emptyset, B)$ using the cumulants of X.
5. Combine to obtain $G = (V, E, H)$.

The cyclic case

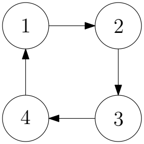
[Joint work *in progress* with Mathias Drton, Marina Garrote-Lopez, and Niko Nikov]

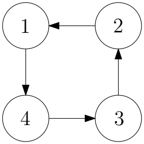


- Can we still uniquely learn the graph?
- What algorithms can we use?

Equivalence classes

- If there are cycles, we cannot learn the graph uniquely.





$$X_1 = \lambda_{41} X_4 + \varepsilon_1,$$

$$X_2 = \lambda_{12} X_1 + \varepsilon_2,$$

$$X_3 = \lambda_{23} X_2 + \varepsilon_3,$$

$$X_4 = \lambda_{34} X_3 + \varepsilon_4.$$

$$X_4 = \frac{1}{\lambda_{41}} X_1 - \frac{1}{\lambda_{41}} \varepsilon_1,$$

$$X_1 = \frac{1}{\lambda_{12}} X_2 - \frac{1}{\lambda_{12}} \varepsilon_2,$$

$$X_2 = \frac{1}{\lambda_{23}} X_3 - \frac{1}{\lambda_{23}} \varepsilon_3,$$

$$X_3 = \frac{1}{\lambda_{34}} X_4 - \frac{1}{\lambda_{34}} \varepsilon_4.$$

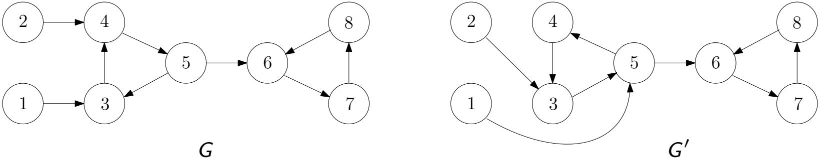
Equivalence classes

Theorem (DGN**R** 2023+)

Two directed graphs G and G' give rise to the same linear non-Gaussian model if and only if there exist vertex-disjoint cycles $C_1, \ldots, C_k$ in G such that

- *the directions of the cycles $C_1, \ldots, C_k$ are reversed in G', and*
- *an edge $v_i \to v_j$ where $v_i \notin C_s$ and $v_j \in C_s$ is in G if and only if $v_i \to v_{j-1}$ is in G', where $v_{j-1} \to v_j$ is on the cycle C_s in G.*

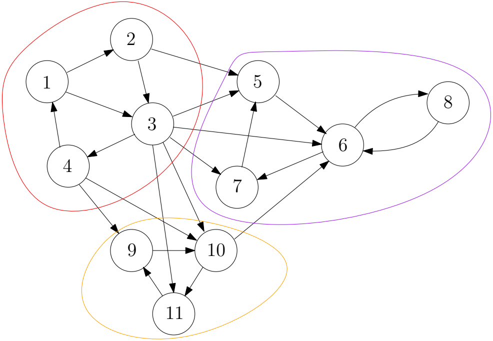
Example



Algorithms for learning the graph (and its equivalence class)

- ICA-based such as [Lacerda, Spirtes, Ramsey, Hoyer, 2012].
- Can we devise a method like [Wang and Drton, 2018]'s for the cyclic case which will also work in the high-dimensional setting?

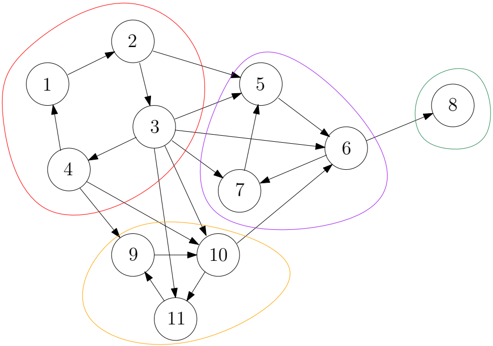
Cyclic graphs



- Strong components: maximal sets of vertices with a directed path between any two of them.
- Get a topological ordering of the strong components
- $\{1, 2, 3, 4\}, \{9, 10, 11\}, \{5, 6, 7, 8\}$.

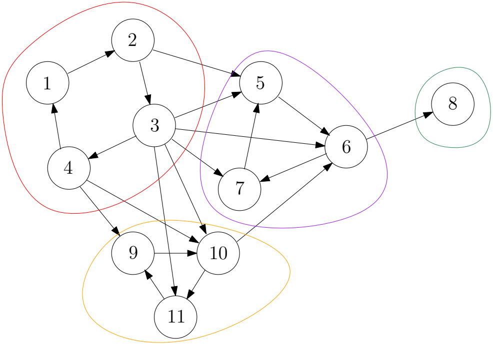
A specific family of cyclic graphs

- We will only consider graphs whose strong components are simple cycles.



A specific family of cyclic graphs

- We will only consider graphs whose strong components are simple cycles.



- Note: it is easy to describe all equivalent graphs to such a graph.

Proposed recovery algorithm: learning the cycles

In: A random vector on n components with 2nd and 3rd moments S and T as above

Out: A causal graph $G = (V, E)$, a representative of an equivalence class

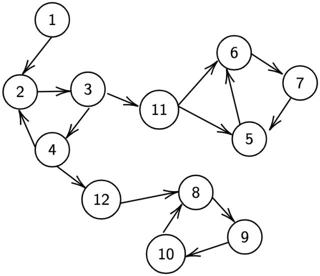
Proposed recovery algorithm: learning the cycles

In: A random vector on n components with 2nd and 3rd moments S and T as above

Out: A causal graph $G = (V, E)$, a representative of an equivalence class

Define for any $u, v \in V$:

$$d_{uv}^{2\times 2} = \det \begin{pmatrix} s_{uu} & s_{uv} \\ t_{uuu} & t_{uuv} \end{pmatrix}, \qquad d_{uv}^{3\times 3} = \det \begin{pmatrix} s_{uu} & s_{uv} & s_{vv} \\ t_{uuu} & t_{uuv} & t_{uvv} \\ t_{uuv} & t_{uvv} & t_{vvv} \end{pmatrix}$$



Proposed recovery algorithm: learning the cycles

In: A random vector on n components with 2nd and 3rd moments S and T as above

Out: A causal graph $G = (V, E)$, a representative of an equivalence class

Define for any $u, v \in V$:

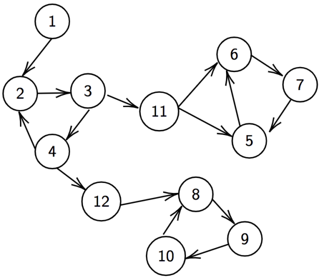
$$d_{uv}^{2\times 2} = \det \begin{pmatrix} s_{uu} & s_{uv} \\ t_{uuu} & t_{uuv} \end{pmatrix}, \qquad d_{uv}^{3\times 3} = \det \begin{pmatrix} s_{uu} & s_{uv} & s_{vv} \\ t_{uuu} & t_{uuv} & t_{uvv} \\ t_{uuv} & t_{uvv} & t_{vvv} \end{pmatrix}$$

Step 1: Compute $d_{uv}^{2\times 2}, d_{uv}^{3\times 3}$ for all $u, v \in V$.

Step 2:
$\mathcal{C}_1 = \{r \in V : d_{rv}^{2\times 2} = d_{rv}^{3\times 3} = 0 \text{ for all } v \in V\}$.

Lemma: $\mathcal{C}_1$ consists of all root nodes.

Step 3: Regress $V \setminus \mathcal{C}_1$ on $\mathcal{C}_1$. Return to Step 1 and repeat Steps 1-3 until $\mathcal{C}_1$ is empty.



Proposed recovery algorithm: learning the cycles

In: A random vector on n components with 2nd and 3rd moments S and T as above

Out: A causal graph $G = (V, E)$, a representative of an equivalence class

Define for any $u, v \in V$:

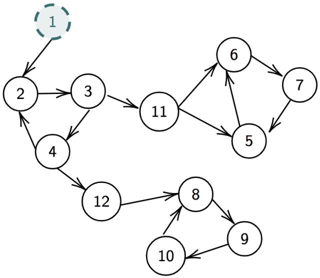
$$d_{uv}^{2\times 2} = \det \begin{pmatrix} s_{uu} & s_{uv} \\ t_{uuu} & t_{uuv} \end{pmatrix}, \qquad d_{uv}^{3\times 3} = \det \begin{pmatrix} s_{uu} & s_{uv} & s_{vv} \\ t_{uuu} & t_{uuv} & t_{uvv} \\ t_{uuv} & t_{uvv} & t_{vvv} \end{pmatrix}$$

Step 1: Compute $d_{uv}^{2\times 2}, d_{uv}^{3\times 3}$ for all $u, v \in V$.

Step 2:
$\mathcal{C}_1 = \{r \in V : d_{rv}^{2\times 2} = d_{rv}^{3\times 3} = 0 \text{ for all } v \in V\}$.

Lemma: $\mathcal{C}_1$ consists of all root nodes.

Step 3: Regress $V \setminus \mathcal{C}_1$ on $\mathcal{C}_1$. Return to Step 1 and repeat Steps 1-3 until $\mathcal{C}_1$ is empty.



Proposed recovery algorithm: learning the cycles

In: A random vector on n components with 2nd and 3rd moments S and T as above

Out: A causal graph $G = (V, E)$, a representative of an equivalence class

Define for any $u, v \in V$:

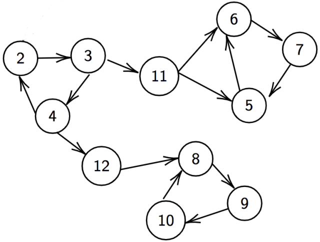
$$d_{uv}^{2\times 2} = \det \begin{pmatrix} s_{uu} & s_{uv} \\ t_{uuu} & t_{uuv} \end{pmatrix}, \qquad d_{uv}^{3\times 3} = \det \begin{pmatrix} s_{uu} & s_{uv} & s_{vv} \\ t_{uuu} & t_{uuv} & t_{uvv} \\ t_{uuv} & t_{uvv} & t_{vvv} \end{pmatrix}$$

Step 1: Compute $d_{uv}^{2\times 2}, d_{uv}^{3\times 3}$ for all $u, v \in V$.

Step 2:
$\mathcal{C}_1 = \{r \in V : d_{rv}^{2\times 2} = d_{rv}^{3\times 3} = 0 \text{ for all } v \in V\}$.

Lemma: $\mathcal{C}_1$ consists of all root nodes.

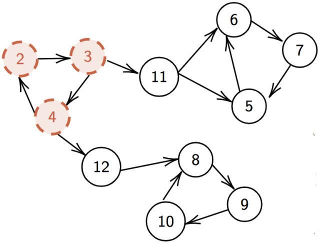
Step 3: Regress $V \setminus \mathcal{C}_1$ on $\mathcal{C}_1$. Return to Step 1 and repeat Steps 1-3 until $\mathcal{C}_1$ is empty.



Proposed recovery algorithm: learning the cycles

Step 4: $\mathcal{C}_2'$ = collection of all maximal $C \subseteq V$ such that $d_{uv}^{3\times 3} = 0$, $d_{uv}^{2\times 2} \neq 0$ for all $u, v \in C$. Prune $\mathcal{C}_2'$ to obtain $\mathcal{C}_2$.

Lemma: $\mathcal{C}_2$ consists of all root cycles.

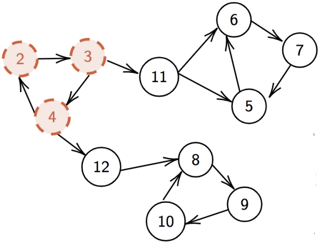


Proposed recovery algorithm: learning the cycles

Step 4: $\mathcal{C}_2'$ = collection of all maximal $C \subseteq V$ such that $d_{uv}^{3\times 3} = 0$, $d_{uv}^{2\times 2} \neq 0$ for all $u, v \in C$. Prune $\mathcal{C}_2'$ to obtain $\mathcal{C}_2$.

Lemma: $\mathcal{C}_2$ consists of all root cycles.

Step 5: Regress the rest of the vertices on $\mathcal{C}_2$. Return to Step 2.

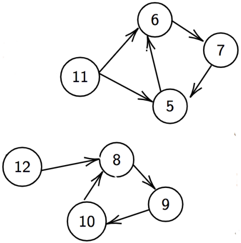


Proposed recovery algorithm: learning the cycles

Step 4: $\mathcal{C}_2'$ = collection of all maximal $C \subseteq V$ such that $d_{uv}^{3\times 3} = 0$, $d_{uv}^{2\times 2} \neq 0$ for all $u, v \in C$. Prune $\mathcal{C}_2'$ to obtain $\mathcal{C}_2$.

Lemma: $\mathcal{C}_2$ consists of all root cycles.

Step 5: Regress the rest of the vertices on $\mathcal{C}_2$. Return to Step 2.

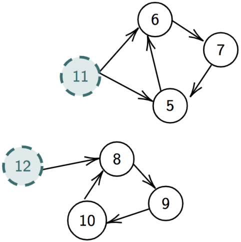


Proposed recovery algorithm: learning the cycles

Step 4: $\mathcal{C}_2' =$ collection of all maximal $C \subseteq V$ such that $d_{uv}^{3\times 3} = 0$, $d_{uv}^{2\times 2} \neq 0$ for all $u, v \in C$. Prune $\mathcal{C}_2'$ to obtain $\mathcal{C}_2$.

Lemma: $\mathcal{C}_2$ consists of all root cycles.

Step 5: Regress the rest of the vertices on $\mathcal{C}_2$. Return to Step 2.

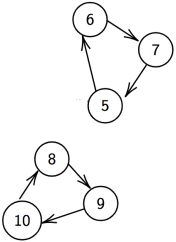


Proposed recovery algorithm: learning the cycles

Step 4: $\mathcal{C}_2'$ = collection of all maximal $C \subseteq V$ such that $d_{uv}^{3\times 3} = 0$, $d_{uv}^{2\times 2} \neq 0$ for all $u, v \in C$. Prune $\mathcal{C}_2'$ to obtain $\mathcal{C}_2$.

Lemma: $\mathcal{C}_2$ consists of all root cycles.

Step 5: Regress the rest of the vertices on $\mathcal{C}_2$. Return to Step 2.

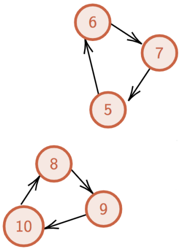


Proposed recovery algorithm: learning the cycles

Step 4: $\mathcal{C}_2'$ = collection of all maximal $C \subseteq V$ such that $d_{uv}^{3\times 3} = 0$, $d_{uv}^{2\times 2} \neq 0$ for all $u, v \in C$. Prune $\mathcal{C}_2'$ to obtain $\mathcal{C}_2$.

Lemma: $\mathcal{C}_2$ consists of all root cycles.

Step 5: Regress the rest of the vertices on $\mathcal{C}_2$. Return to Step 2.

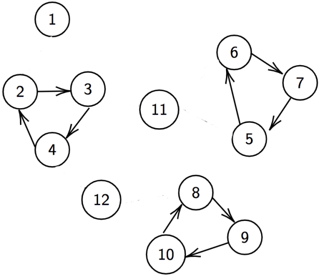


Proposed recovery algorithm

We know: the cycles and a topological ordering:

$$\{1\}, \{2, 3, 4\}, \{11\}, \{12\}, \{5, 6, 7\}, \{8, 9, 10\}.$$

Step 6: Learn ancestry relationships among cycles: use regression backwards in topological order.

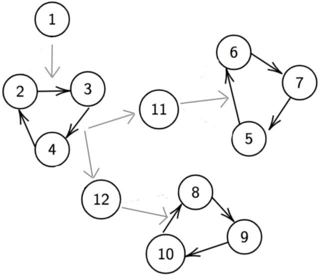


Proposed recovery algorithm

We know: the cycles and a topological ordering:

$$\{1\}, \{2, 3, 4\}, \{11\}, \{12\}, \{5, 6, 7\}, \{8, 9, 10\}.$$

Step 6: Learn ancestry relationships among cycles: use regression backwards in topological order.



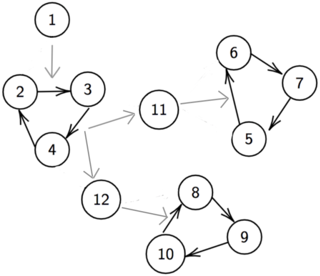
Proposed recovery algorithm

We know: the cycles and a topological ordering:

$$\{1\}, \{2, 3, 4\}, \{11\}, \{12\}, \{5, 6, 7\}, \{8, 9, 10\}.$$

Step 6: Learn ancestry relationships among cycles: use regression backwards in topological order.

Step 7: Starting from the source nodes/cycles, learn the edges and their weights.



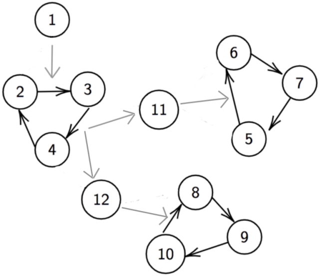
Proposed recovery algorithm

We know: the cycles and a topological ordering:

$$\{1\}, \{2, 3, 4\}, \{11\}, \{12\}, \{5, 6, 7\}, \{8, 9, 10\}.$$

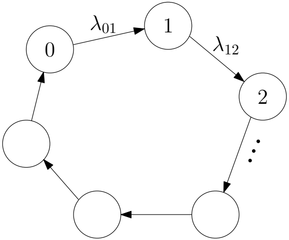
Step 6: Learn ancestry relationships among cycles: use regression backwards in topological order.

Step 7: Starting from the source nodes/cycles, learn the edges and their weights.



- Learn the skeleton for each root cycle by using conditional independence.
- Turn all root cycles into root nodes by "undoing them".
- Learn edges/weights between cycles.

"Undoing" a root cycle



Lemma:

- If the cycle length is at least 3, then there is a linear equation in λ_{01}

$$p(s_{ij}, t_{ijk} : i, j, k \in \{0, 1, 2\})\lambda_{01} = q(s_{ij}, t_{ijk} : i, j, k \in \{0, 1, 2\}),$$

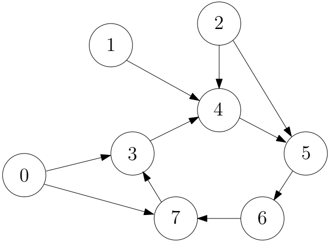
  where $p(s_{ij}, t_{ijk} : i, j, k \in \{0, 1, 2\})$ is nonzero with probability 1. Thus, we can compute λ_{01} uniquely.

- If the cycle length is 2, there is a quadratic equation

$$p(s_{ij}, t_{ijk})\lambda_{01}^2 + q(s_{ij}, t_{ijk})\lambda_{0,1} + r(s_{ij,t_{ijk}}) = 0.$$

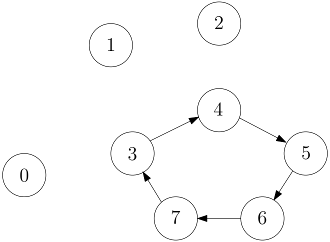
  with $p(s_{ij}, t_{ijk})$ nonzero. Thus, there are two solutions for λ_{01}.

Learning edges/weights between cycles



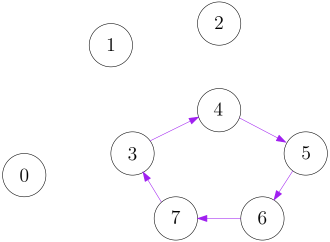
- Regress on all parent nodes and take residuals.
- Learn edge weights on cycle.
- From regression coefficients, learn edge weights from parent nodes to cycle.

Learning edges/weights between cycles



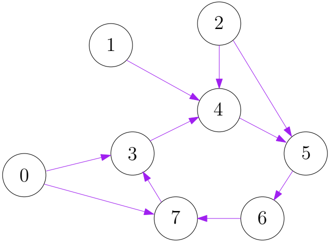
- Regress on all parent nodes and take residuals.
- Learn edge weights on cycle.
- From regression coefficients, learn edge weights from parent nodes to cycle.

Learning edges/weights between cycles



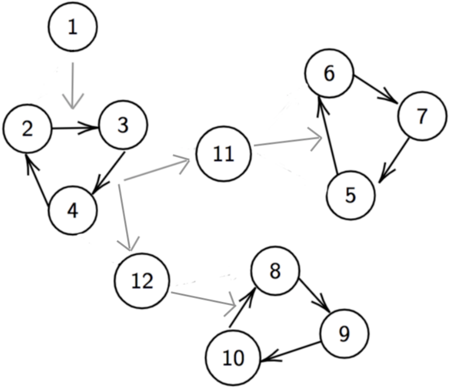
- Regress on all parent nodes and take residuals.
- Learn edge weights on cycle.
- From regression coefficients, learn edge weights from parent nodes to cycle.

Learning edges/weights between cycles

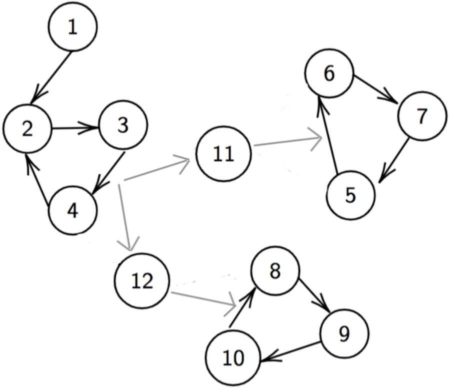


- Regress on all (possible) parent nodes and take residuals.
- Learn edge weights on cycle.
- From regression coefficients, learn edge weights from parent nodes to cycle.

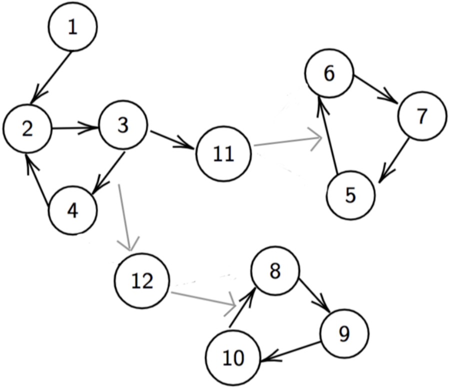
Proposed algorithm: learning edges and edge weights



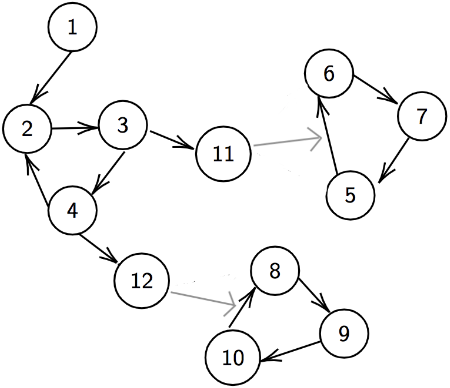
Proposed algorithm: learning edges and edge weights



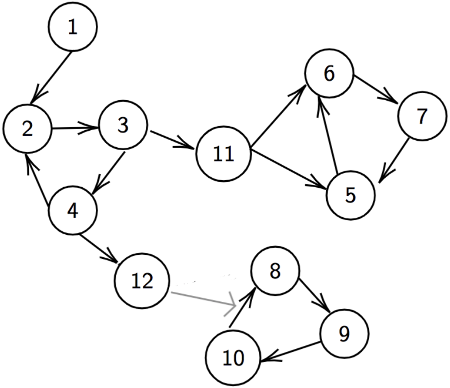
Proposed algorithm: learning edges and edge weights



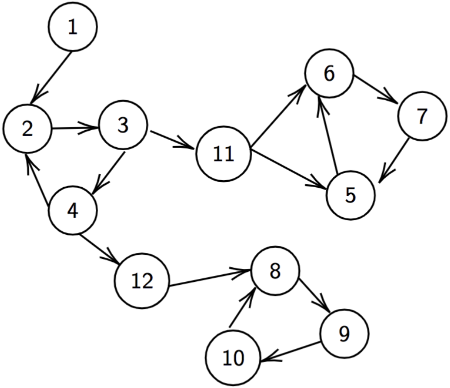
Proposed algorithm: learning edges and edge weights



Proposed algorithm: learning edges and edge weights



Proposed algorithm: learning edges and edge weights



Further questions

- Implement algorithm and compare with others
- Compute sample size complexity
- Extend to all cyclic graphs
- More algebraic constraints that can be used?

Thank you!

C. Améndola, M. Drton, A. Grosdos, R. Homs-Pons, and E. Robeva. *Third-order moment varieties of non-Gaussian graphical models*. Information and Inference (2023)

M. Drton, M. Garrote-Lopéz, N. Nikov, and E. Robeva. *Learning cyclic linear non-Gaussian causal models via algebraic constraints*. In preparation

Y. Liu, E. Robeva, and H. Wang. *Learning Linear Non-Gaussian Graphical Models with Multidirected Edges*. Journal of Causal Inference (2021)

E. Robeva and J.B. Seby. *Multi-trek Separation in Linear Structural Equation Models*. SIAM Journal on Applied Algebra and Geometry (SIAGA) (2021)

More algebraic constraints

- (Robeva, Seby, 2020): Characterize vanishing of *determinants of subtensors* of k-th cumulant tensor $C^{(k)}$ in a LiNGAM with graph $G = (V, E, H)$;

$$\det(C^{(k)}_{A_1,\dots,A_k}) = 0$$

if and only if every system of k-treks between $A_1, \dots, A_k$ has a sided intersection.

Here:

$$\det(T) = \sum_{\sigma_2,\dots,\sigma_k \in \mathfrak{S}(d)} \operatorname{sign}(\sigma_2) \cdots \operatorname{sign}(\sigma_k) \prod_{i=1}^{d} T_{i,\sigma_2(i),\dots,\sigma_k(i)}$$

is the combinatorial hyperdeterminant.

More algebraic constraints

- (Robeva, Seby, 2020): Characterize vanishing of *determinants of subtensors* of k-th cumulant tensor $C^{(k)}$ in a LiNGAM with graph $G = (V, E, H)$;

  $$\det(C^{(k)}_{A_1,\ldots,A_k}) = 0$$

  if and only if every system of k-treks between $A_1, \ldots, A_k$ has a sided intersection.

  Here:

  $$\det(T) = \sum_{\sigma_2,\ldots,\sigma_k \in \mathfrak{S}(d)} \text{sign}(\sigma_2) \cdots \text{sign}(\sigma_k) \prod_{i=1}^{d} T_{i,\sigma_2(i),\ldots,\sigma_k(i)}$$

  is the combinatorial hyperdeterminant.

- Can we learn such relationships in the case of both cycles and hidden variables?