Graphical Models For Complex Health Data (P8124)

Daniel Malinsky

Columbia University
dsm2128@cumc.columbia.edu

Structure Learning (part 2)

Recap/overview

We've been operating in a setting with little background knowledge: we have a set of random variables, and some broad background assumptions about the class of models that might have generated the data (e.g., a DAG or UG). Our task has been to try to learn the graphical structure from data. Why would we do this?

- We might be interested in the (conditional) (in)dependence relations represented by a graph
  - May help answer questions like "what are good predictors of some target Y?" "what is the Markov blanket of Y?" "is X_i still correlated with (dependent on) Y if I control for X_j"?
  - May subsequently estimate parameters in a model satisfying some independence constraints using MLE etc
- We might be interested in using exact or approximate inference methods to compute some marginal/conditional distributions, or MAP estimates (tbd later)

Recap/overview

We introduced two broad strategies for structure learning: constraint-based and score-based.

These methods were originally developed for the purposes of *causal structure learning* (a.k.a. "causal discovery").

- We might be interested in the causal pathways and structural features represented by a graph
  - May help answer questions like "what are the direct causes or causal ancestors of Y?" "is the effect of X_i on Y mediated through X_j"? etc.
- We might be interested in estimating causal effects or post-intervention distributions, for which knowing the graph is helpful (backdoor criterion, etc)

Latent variables

In reality, the "true" causal process probably includes a bunch of variables not represented in our data. Unmeasured variables are called "latent" or "hidden" and these pose a real problem for causal inference and causal learning.

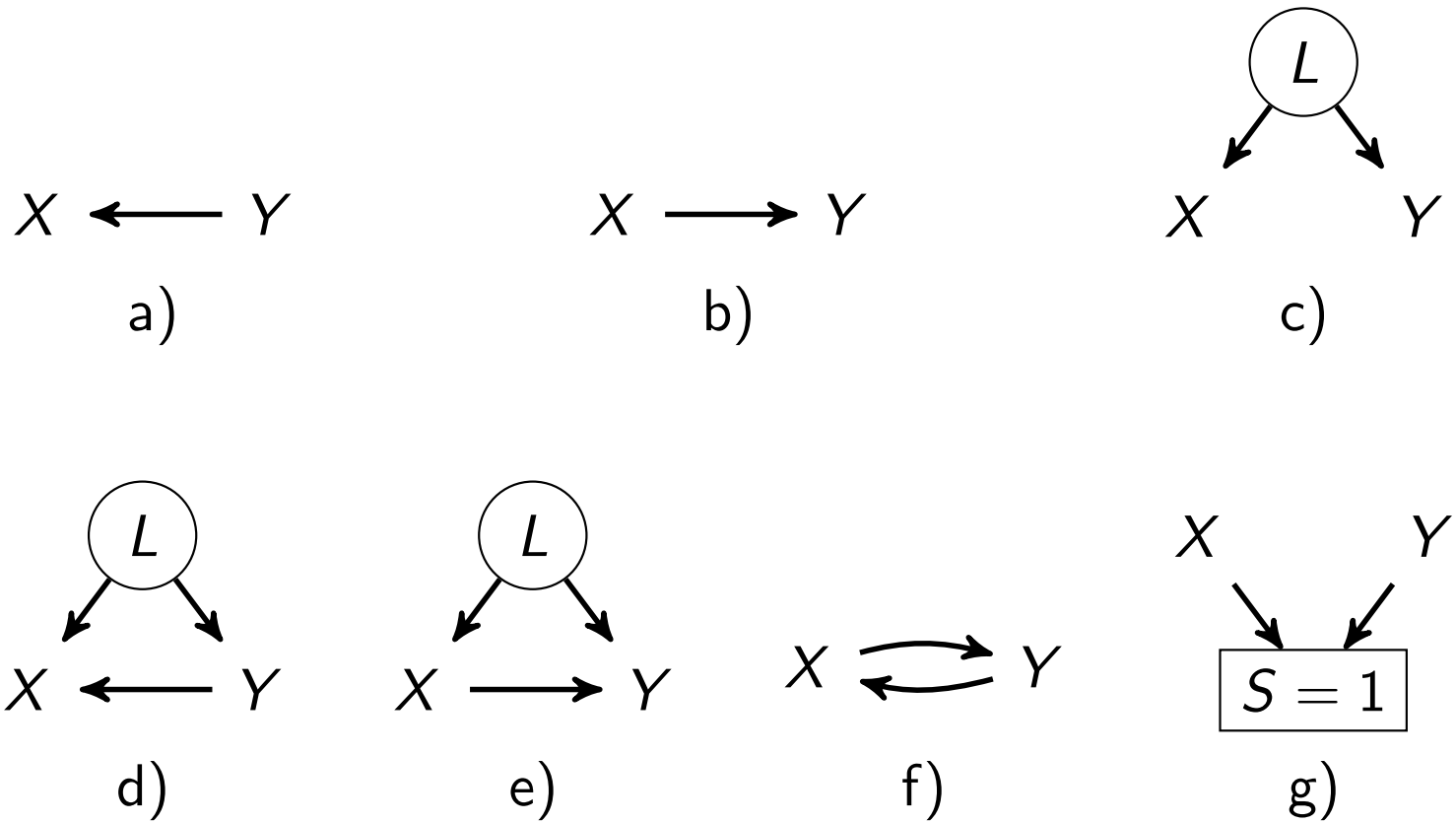
Latent variables

In reality, the "true" causal process probably includes a bunch of variables not represented in our data. Unmeasured variables are called "latent" or "hidden" and these pose a real problem for causal inference and causal learning.

For example, the underlying causal process may be described by a DAG $\mathcal{G} = (V, E)$ with vertices $V = X \cup L$, but we only observe X.

Latent variables from a structure learning point-of-view

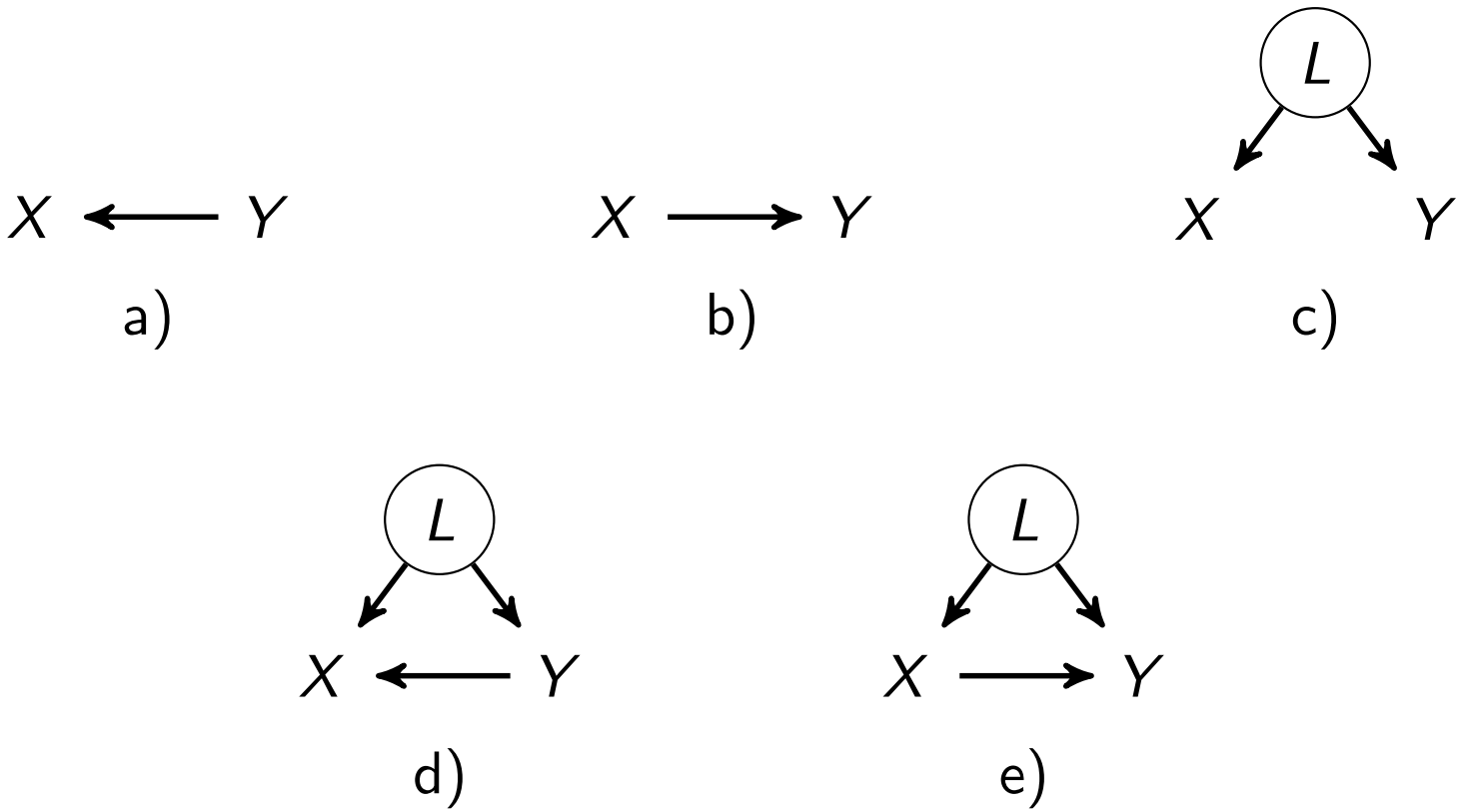
Consider two observed variables X and Y which are known to be dependent. What causal processes may *explain* this dependence?



+ combinations of f) & g) with the others.

Latent variables from a structure learning point-of-view

Consider two observed variables X and Y which are judged to be dependent. What causal processes may *explain* this dependence? (Let's exclude feedback and selection bias for the time being.)



How could we possibly distinguish between these possibilities from (observed) conditional (in)dependence facts alone?

Distinguishing "real" causality from latent confounding

In general, with just two variables (+ no background knowledge about the latents, no other assumptions) we cannot distinguish between those possibilities. They all imply the same restrictions on the distribution $p(x, y)$: i.e., no restrictions at all.

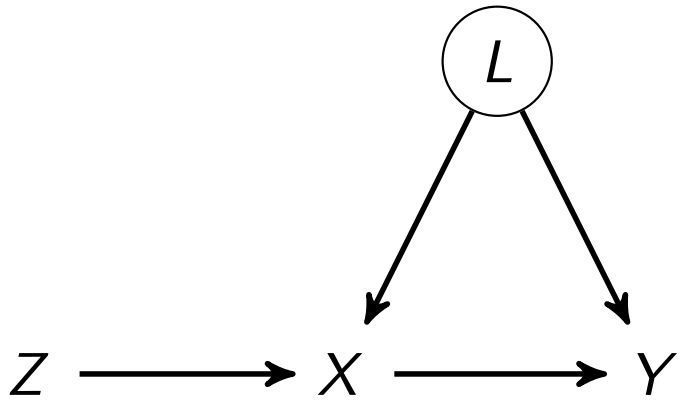
However, with > 2 variables, some *patterns of independence* may help narrow down the structure (assuming faithfulness).

Distinguishing "real" causality from latent confounding

In general, with just two variables (+ no background knowledge about the latents, no other assumptions) we cannot distinguish between those possibilities. They all imply the same restrictions on the distribution $p(x, y)$: i.e., no restrictions at all.

However, with > 2 variables, some *patterns of independence* may help narrow down the structure (assuming faithfulness).

For example:



$\Rightarrow$ Y and Z are **not** independent given X. (Why?)

Distinguishing "real" causality from latent confounding

In general, with just two variables (+ no background knowledge about the latents, no other assumptions) we cannot distinguish between those possibilities. They all imply the same restrictions on the distribution $p(x, y)$: i.e., no restrictions at all.

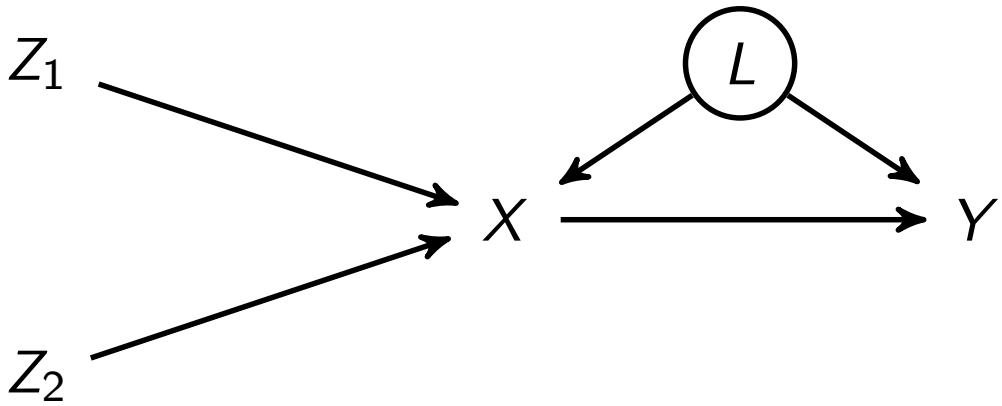
However, with > 2 variables, some *patterns of independence* may help narrow down the structure (assuming faithfulness).

For example:

$$Z \longrightarrow X \longrightarrow Y$$

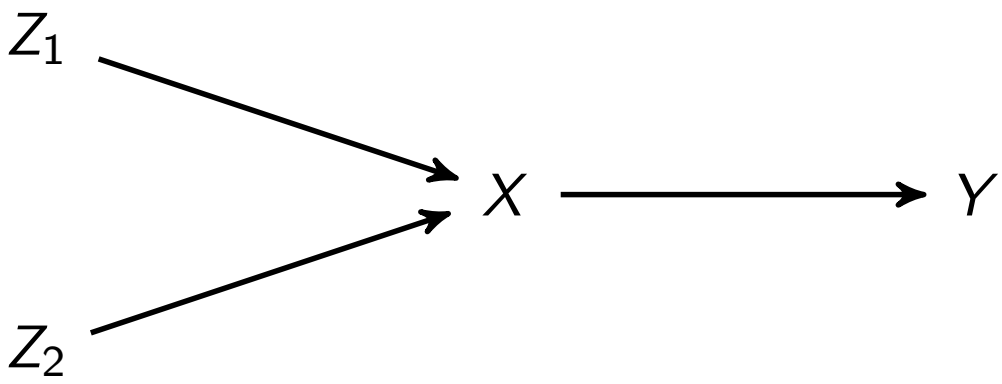
$\Rightarrow$ Y and Z **are** independent given X. (Why?)

Patterns of independence constraints may rule out latent confounding



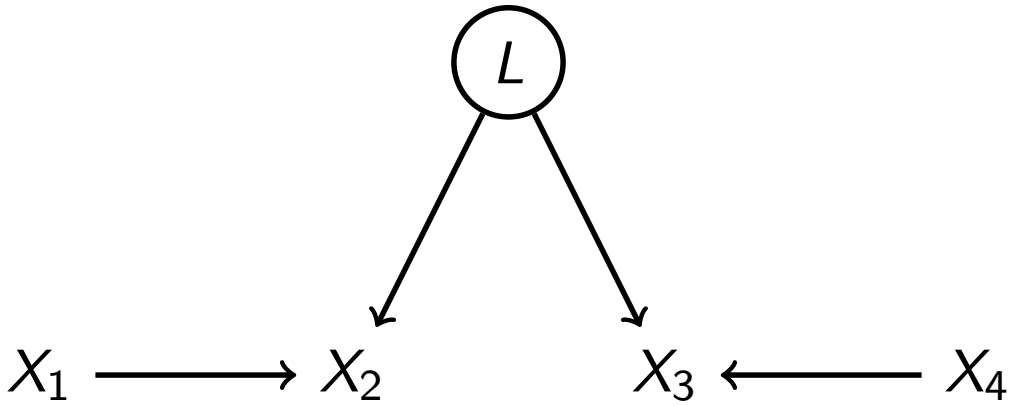
$Z_1 \perp\!\!\!\perp Z_2$
$Z_1 \not\perp\!\!\!\perp Z_2 | X$
$Y \not\perp\!\!\!\perp \{Z_1, Z_2\}$
$Y \not\perp\!\!\!\perp Z_1 | X$
$Y \not\perp\!\!\!\perp Z_2 | X$

Patterns of independence constraints may rule out latent confounding



$Z_1 \perp\!\!\!\perp Z_2$
$Z_1 \not\perp\!\!\!\perp Z_2 | X$
$Y \not\perp\!\!\!\perp \{Z_1, Z_2\}$
$Y \perp\!\!\!\perp Z_1 | X$
$Y \perp\!\!\!\perp Z_2 | X$

Patterns of independence constraints may also *suggest* latent confounding



$X_1 \not\perp\!\!\!\perp X_2$ and $X_2 \not\perp\!\!\!\perp X_3$ and $X_3 \not\perp\!\!\!\perp X_4$
$X_1 \perp\!\!\!\perp X_4$ and $X_1 \perp\!\!\!\perp X_3$ and $X_2 \perp\!\!\!\perp X_4$
$X_1 \not\perp\!\!\!\perp X_3 | X_2$
$X_2 \not\perp\!\!\!\perp X_4 | X_3$

Patterns of independence constraints may also *suggest* latent confounding

$X_1 \not\perp\!\!\!\perp X_2$ and $X_2 \not\perp\!\!\!\perp X_3$ and $X_3 \not\perp\!\!\!\perp X_4$
$X_1 \perp\!\!\!\perp X_4$ and $X_1 \perp\!\!\!\perp X_3$ and $X_2 \perp\!\!\!\perp X_4$
$X_1 \not\perp\!\!\!\perp X_3 | X_2$
$X_2 \not\perp\!\!\!\perp X_4 | X_3$

$\Rightarrow$ recall ADMGs from many lectures ago!

How can we generalize these ideas to construct something like the PC algorithm for settings with latent variables?

First, consider what we'd have to assume to interpret the output of algorithms such as PC & GES *causally*.
Input: data over X
Output: a CPDAG over X

To understand this as a causal discovery procedure we must assume that the underlying DGP is a (unknown) causal DAG over the variables X, and that the distribution $p(x)$ is Markov and faithful to that causal DAG. This means assuming that X contains all the "causally relevant" variables, i.e., that there are no unmeasured common causes of any two variables in X ("causal sufficiency" or "no latent confounders").

If that assumption is correct then algorithms like PC and GES may learn the structure of that DGP (up to Markov equivalence). However, if there are latent confounders, then PC and GES will produce causally incorrect conclusions.

Constraint-based structure learning in the presence of latent variables

The assumption of causal sufficiency is rarely warranted in practice! Fortunately, there exist procedures which perform structure learning which dispense with the causal sufficiency assumption, and allow for arbitrary latent variables. One constraint-based procedure, which follows similar logic to PC, is called the FCI (Fast Causal Inference) algorithm.

We don't want to perform search for the best DAG (or CPDAG) which "fits" the data, since in general no DAG over X will do. We have to consider searching over a different space of graphs.

Latent projections

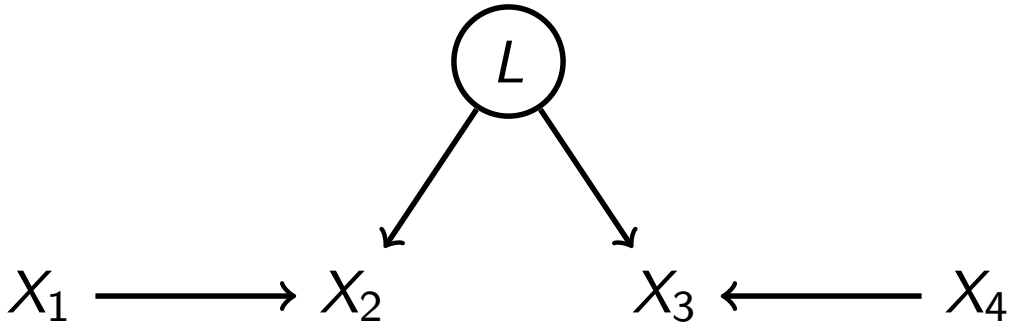
Beginning with a DAG $\mathcal{G}$, it is useful to think about the induced dependence structure when some variables have been marginalized out. The conditional independence relations in the *marginal* distribution are represented by an ADMG. One may construct an ADMG via the operation of *latent projection*.

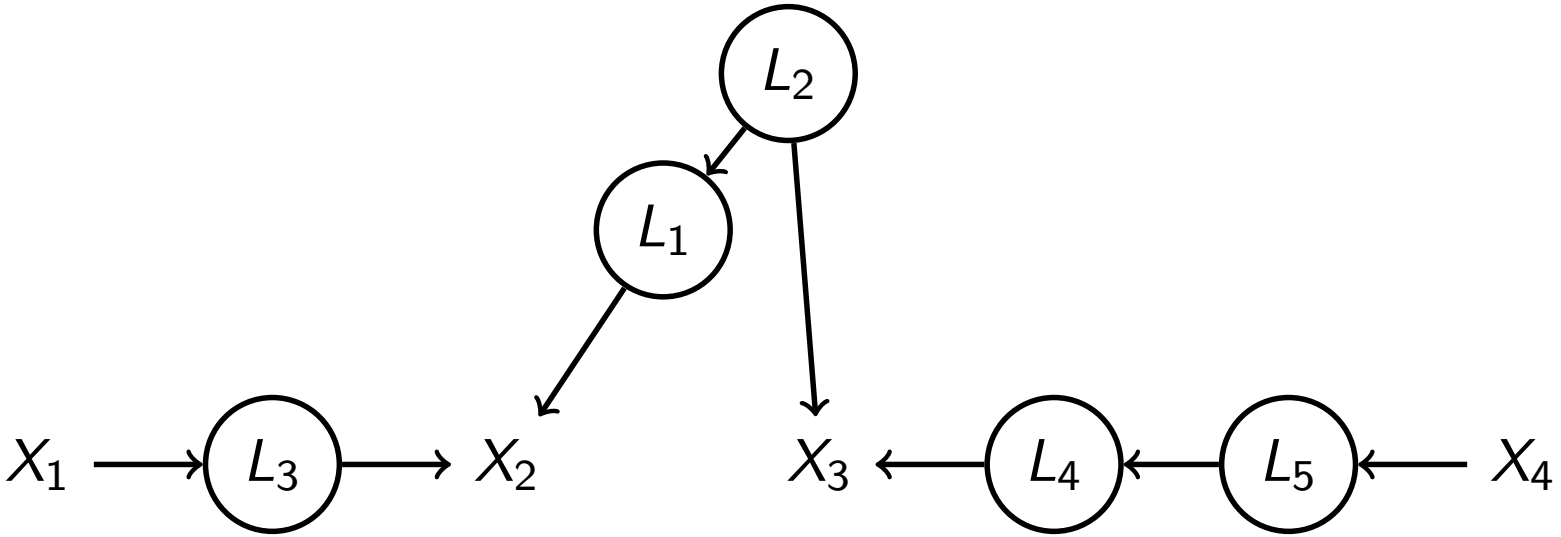
Consider a DAG $\mathcal{G} = (V, E)$ with vertex set $V = X \cup L$. The latent projection $\mathcal{G}' = (V', E')$ is a mixed graph with vertex set $V' = X$ such that:

- for any $X_i, X_j \in X$ there is an edge $X_i \to X_j$ if there exists a directed path from X_i to X_j in $\mathcal{G}$, with all intermediate nodes on the path in L
- there is an edge $X_i \leftrightarrow X_j$ if there exists a path from X_i to X_j of the form $X_i \leftarrow \cdots \rightarrow X_j$, where every intermediate node on the path is in L and no consecutive edges on the path are of the form $\rightarrow L_k \leftarrow$ for $L_k \in L$.

Latent projections

Note that an infinite number of distinct latent variable DAGs will share the same latent projection ADMG!





ADMGs

ADMGs preserve conditional independence relations among the *observed* variables. There is a separation criterion which generalizes d-separation to structures with bidirected edges: m-separation. For A, B, C disjoint subsets of X: $A \perp_d^{\mathcal{G}} B|C \iff A \perp_m^{\mathcal{G}'} B|C$.

There's also a lot of well-developed theory about deriving causal effects or post-intervention distributions (if possible) given an ADMG. We won't go into that here.[1]

[1] See the review article by Shpitser (2018), "Identification in causal graphical models" in *Handbook of Graphical Models*, CRC Press.

Direct structure learning of ADMGs is hard, not well-developed

Though ADMGs have a relatively clear causal interpretation, some nice properties, and lots of associated theory, they are not good targets for structure learning. Why?

- Markov equivalence for ADMGs is complicated
- Parameterization of ADMGs is complicated (except for binary variables)
- ADMGs are not *maximal*

Maximality

Def. A graph $\mathcal{G}$ is said to be *maximal* if for every pair of vertices X_i, X_j

$$X_i \not\sim X_j \implies \exists S \subseteq V \setminus \{X_i, X_j\} \text{ such that } X_i \perp\!\!\!\perp X_j | X_S$$

.

Thus a graph is maximal if every missing edge corresponds to at least one independence in the model. No additional edge may be added to a maximal graph without changing the independence model. (DAGs and UGs are maximal. ADMGs are not.)

Maximal Ancestral Graphs

To make a PC-style search procedure possible, we focus on a class of graphs called Maximal Ancestral Graphs (MAGs). For the purposes of the present discussion, we can view MAGs as a special type of ADMG.

Def. If $X_i \leftrightarrow X_j$ in $\mathcal{G}$ then $X_i \in \text{Sp}(X_j, \mathcal{G})$.

Def. A (directed)[2] ancestral graph $\mathcal{G}$ is a mixed graph ($\rightarrow$ and $\leftrightarrow$ edges) such that $\forall X_i \in X$, $X_i \notin \text{An}(\text{Pa}(X_i, \mathcal{G}) \cup \text{Sp}(X_i, \mathcal{G}))$. That is, an ancestral graph does not contain any directed or almost directed cycles.

Def. A MAG is an ancestral graph which is maximal.

[2] MAGs are actually more general than this: they can have undirected ($-$) edges to represent selection bias in addition to latent confounding, but I'm going to ignore that in this presentation.

Maximal Ancestral Graphs

MAGs have some pros and cons.

Pros:

- They are maximal, so if we find a conditional independence we can remove an edge in a PC-style search.
- We can characterize Markov equivalent MAGs.
- They have other "nice" properties of ADMGs, m-separation works out, etc.

Cons:

- They have a somewhat confusing interpretation!
- Less "informative" than ADMGs

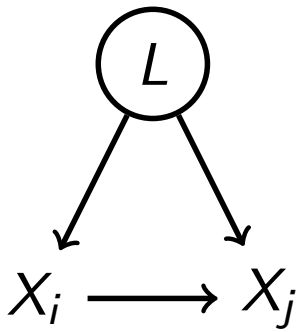
Maximal Ancestral Graphs

$X_i \to X_j$ in a MAG means that X_i is an ancestor of X_j in the underlying DAG $\mathcal{G}$.

$X_i \leftrightarrow X_j$ means that X_i is not an ancestor of X_j and X_j is not an ancestor of X_i, which implies that there is a latent common cause of X_i and X_j in $\mathcal{G}$.

NOTE: An ancestral relationship + latent confounding can coexist! So just because $X_i \to X_j$ in a MAG does not mean there is no latent common cause between X_i and X_j.

$$X_i \longrightarrow X_j \quad \ldots \text{ in a MAG}$$



... may hide in the underlying DAG

Maximal Ancestral Graphs

We can construct a MAG from a DAG by a procedure similar to latent projection.

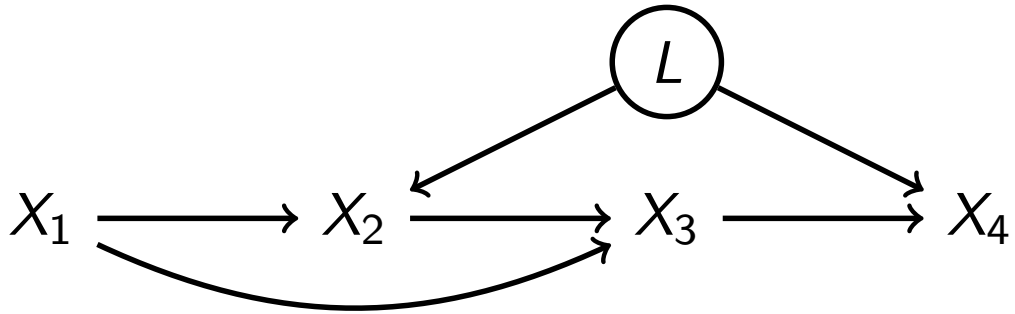
Def. An *inducing path relative to* L is a path on which every vertex not in L (except for the endpoints) is a collider on the path and every collider is an ancestor of an endpoint of the path.

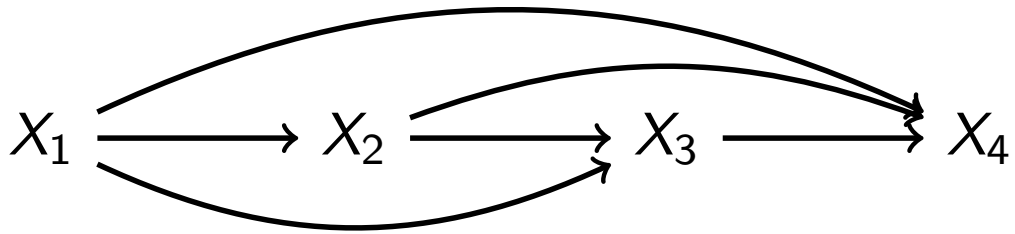
Start with a DAG $\mathcal{G}$ over $X \cup L$ and construct a MAG $\mathcal{G}'$:

- for each pair of variables $X_i, X_j \in X$, X_i and X_j are adjacent in $\mathcal{G}'$ if and only if there is an inducing path between them relative to L in $\mathcal{G}$.
- for each pair of adjacent variables X_i, X_j in $\mathcal{G}'$, orient the edge as $X_i \to X_j$ in $\mathcal{G}'$ if $X_i \in \text{An}(X_j, \mathcal{G})$; orient it as $X_i \leftarrow X_j$ in $\mathcal{G}'$ if $X_j \in \text{An}(X_i, \mathcal{G})$; orient it as $X_i \leftrightarrow X_j$ in $\mathcal{G}'$ otherwise.

Maximal Ancestral Graphs

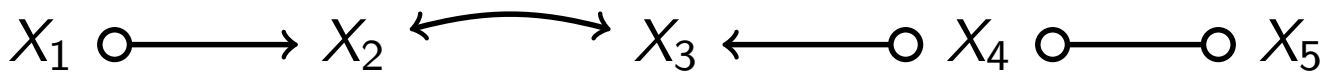
A MAG will have "extra" adjacencies that do not correspond to adjacencies in the underlying DAG. These are adjacencies induced by the latent confounders, and which must be there to preserve maximality.





Partial Ancestral Graphs

A Markov equivalence class of MAGs is represented by a PAG. A PAG is a mixed graph that has $\circ\!\!\rightarrow$ and $\circ\!\!-\!\!\circ$ edges to represent uncertainty about edge endpoints. $\circ$ can correspond to a "tail" or "arrowhead."



Just like a CPDAG represents a set of DAGs, a PAG represents a set of MAGs that each imply the same set of independence constraints.

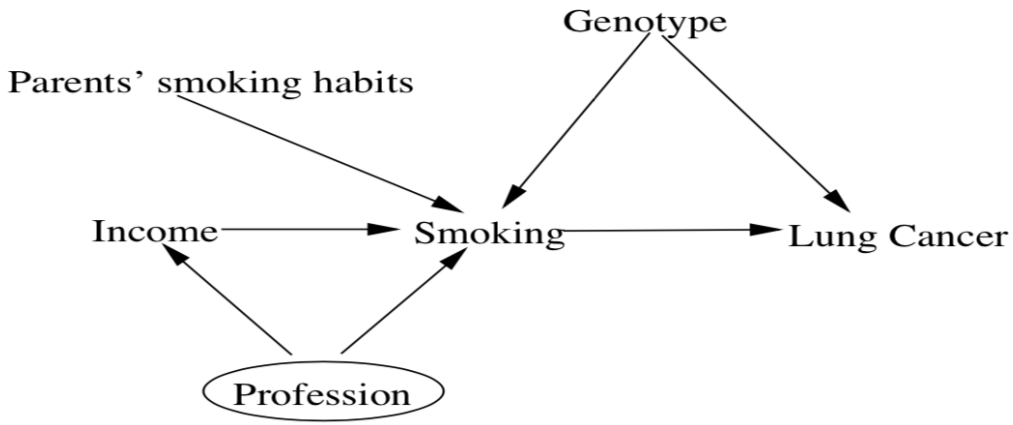


Figure 2: A causal DAG with a latent variable.

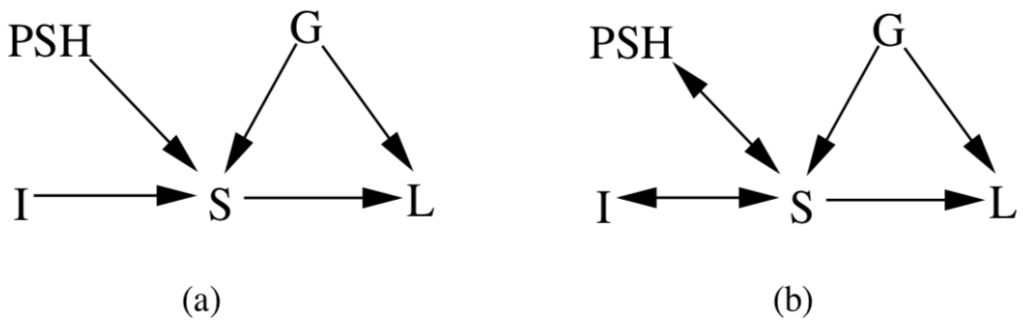


Figure 3: Two Markov Equivalent MAGs.

from Zhang (2008a)

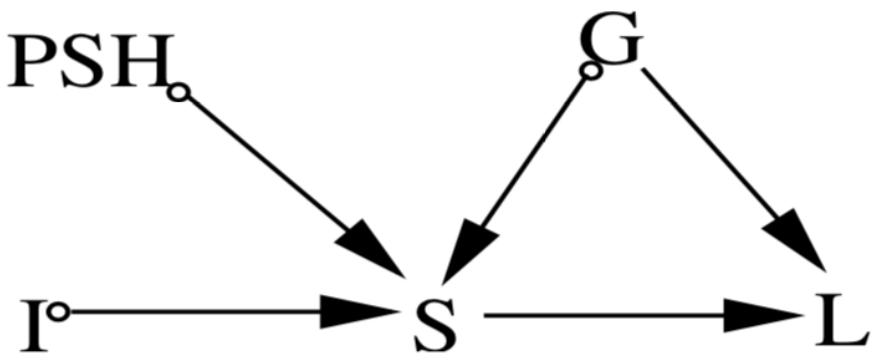


Figure 4: The PAG in our five-variable example.

from Zhang (2008a)

Algorithm 0.1: FCI(TEST, α)

Input: Samples of the vector $X = (X_1, \ldots, X_p)$
Output: PAG $\mathcal{P}$

1. Form the complete graph $\mathcal{P}$ on vertex set X with $\circ\!\!-\!\!\circ$ edges.
2. $s \leftarrow 0$
3. **repeat**
4. **for all** pairs of adjacent vertices (X_i, X_j) s.t. $|\operatorname{Adj}(X_i, \mathcal{P}) \setminus \{X_j\}| \geq s$ and subsets $S \subset \operatorname{Adj}(X_i, \mathcal{P}) \setminus \{X_j\}$ s.t. $|S| = s$
5. **if** $X_i \perp\!\!\!\perp X_j | X_S$ according to (TEST, α) **then** { Delete edge $X_i \circ\!\!-\!\!\circ X_j$ from $\mathcal{P}$. Let $sepset(X_i, X_j) = sepset(X_j, X_i) = S$.
6. **end**
7. $s \leftarrow s + 1$
8. **until** for each pair of adjacent vertices (X_i, X_j), $|\operatorname{Adj}(X_i, \mathcal{P}) \setminus \{X_j\}| < s$.
9. **for all** triples (X_i, X_k, X_j) s.t. $X_i \in \operatorname{Adj}(X_k, \mathcal{P})$ and $X_j \in \operatorname{Adj}(X_k, \mathcal{P})$ but $X_i \notin \operatorname{Adj}(X_j, \mathcal{P})$, orient $X_i \ast\!\!\rightarrow X_k \leftarrow\!\!\ast X_j$ iff $X_k \notin sepset(X_i, X_j)$.
10. **for all** pairs (X_i, X_j) adjacent in $\mathcal{P}$ **if** $\exists S$ s.t. $S \in \operatorname{pds}(X_i, X_j, \mathcal{P})$ or $S \in \operatorname{pds}(X_j, X_i, \mathcal{P})$ and $X_i \perp\!\!\!\perp X_j | X_S$ according to (TEST, α) **then** { Delete edge $X_i \ast\!\!-\!\!\ast X_j$ from $\mathcal{P}$. Let $sepset(X_i, X_j) = sepset(X_j, X_i) = S$.
11. Reorient all edges as $\circ\!\!-\!\!\circ$ and **repeat** step 9.
12. Exhaustively apply orientation rules (R1-R10) in Zhang (2008b) to orient remaining endpoints.
13. **return** $\mathcal{P}$.

Let $X \in \operatorname{pds}(X_i, X_j, \mathcal{G})$ if and only if $X \neq X_i$, $X \neq X_j$, and there is a path π between X_i and X in $\mathcal{G}$ such that for every subpath $\langle X_m, X_l, X_h \rangle$ of π either X_l is a collider on the subpath in $\mathcal{G}$ or $\langle X_m, X_l, X_h \rangle$ is a triangle in $\mathcal{G}$. A *triangle* is a triple $\langle X_m, X_l, X_h \rangle$ where each pair of vertices is adjacent.

Zhang (2008b) refers to "On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias," Artificial Intelligence 172: 1873-1896.

Example in R using pcalg package

Visible edges

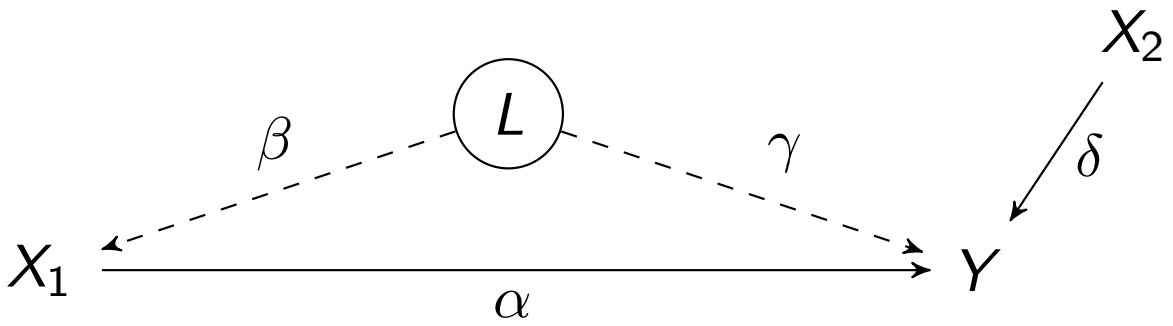
Def. Given a MAG $\mathcal{M}$ / PAG $\mathcal{P}$, a directed edge $X \to Y$ in $\mathcal{M}$ / $\mathcal{P}$ is *visible* if there is a vertex Z not adjacent to Y, such that there is an edge between Z and X that is into X (has an arrowhead at X), or there is a collider path between Z and X that is into X and every non-endpoint vertex on the path is a parent of Y. Otherwise $X \to Y$ is said to be invisible. (All directed edges in a DAG/CPDAG are visible.)

Visible edges

Def. Given a MAG $\mathcal{M}$ / PAG $\mathcal{P}$, a directed edge $X \to Y$ in $\mathcal{M}$ / $\mathcal{P}$ is *visible* if there is a vertex Z not adjacent to Y, such that there is an edge between Z and X that is into X (has an arrowhead at X), or there is a collider path between Z and X that is into X and every non-endpoint vertex on the path is a parent of Y. Otherwise $X \to Y$ is said to be invisible. (All directed edges in a DAG/CPDAG are visible.)

Directed edges that are visible do not "hide" confounders, they correspond to unconfounded causal effects.

Unidentifiable causal effects



Say you know the association between X_1 and Y is θ. How much of this can be "attributed" to α versus β, γ?

Backdoor criterion for MAGs/PAGs

Def. Let X be a vertex in $\mathcal{G}$, where $\mathcal{G}$ represents a causal DAG, CPDAG, MAG, or PAG. Let $\mathcal{R}$ be a DAG or MAG represented by $\mathcal{G}$, in the following sense. If $\mathcal{G}$ is a DAG or MAG, we simply let $\mathcal{R} = \mathcal{G}$. If $\mathcal{G}$ is a CPDAG/PAG, we let $\mathcal{R}$ be a DAG/MAG in the Markov equivalence class described by $\mathcal{G}$ with the same number of edges into X as $\mathcal{G}$. Let $\mathcal{R}_{\underline{X}}$ be the graph obtained from $\mathcal{R}$ by removing all directed edges out of X that are visible in $\mathcal{P}$.

Def. Let X and Y be two distinct vertices in mixed graph $\mathcal{G}$. We say that $V \in \text{Dsep}(X, Y, \mathcal{G})$ if $V \neq X$ and there is a collider path between X and V in $\mathcal{G}$, such that every vertex on this path is an ancestor of X or Y in $\mathcal{G}$.

Def. If there is a possibly directed path from X to Y (or if $X = Y$) then Y is a possible descendent of X. Let $\text{possDe}(X, \mathcal{G})$ denote the set of possible descendents of X.

Backdoor criterion for MAGs/PAGs

Theorem.[3] Let X and Y be two distinct vertices in a causal DAG, CPDAG, MAG, or PAG $\mathcal{G}$. Let $\mathcal{R}$ and $\mathcal{R}_{\underline{X}}$ be defined as above. If $Y \in \text{Adj}(X, \mathcal{R}_{\underline{X}})$ or $\text{Dsep}(X, Y, \mathcal{R}_{\underline{X}}) \cap \text{possDe}(X, \mathcal{G}) \neq \emptyset$, then $p(y|\,\text{do}(x))$ is not identifiable via the generalized backdoor criterion. Otherwise $\text{Dsep}(X, Y, \mathcal{R}_{\underline{X}})$ satisfies the generalized backdoor criterion relative to (X, Y) and $\mathcal{G}$.

That is, when $\text{Dsep}(X, Y)$ satisfies this criterion, it is sufficient to adjust for the variables in $\text{Dsep}(X, Y, \mathcal{R}_{\underline{X}})$ to estimate the causal effect of X on Y.

[3] Maathuis and Colombo (2015). A generalized back-door criterion. Annals of Statistics, 43(3), 1060-1088.

References on MAGs, PAGs, and FCI

Ali, Richardson, and Spirtes (2009) "Markov equivalence or ancestral graphs" Annals of Statistics 37(5B): 2808–2837.

Colombo, Maathuis, Kalisch, and Richardson (2012) "Learning high-dimensional directed acyclic graphs with latent and selection variables," Annals of Statistics 40(1): 294-321.

Maathuis and Colombo (2015). "A generalized back-door criterion." Annals of Statistics, 43(3), 1060-1088.

Richardson and Spirtes (2002) "Ancestral graph Markov models," Annals of Statistics 30(4): 962–1030.

Spirtes, Scheines, and Glymour (2000) Causation, Prediction, and Search, MIT Press.

Zhang (2008a) "Causal reasoning with ancestral graphs," JMLR 9: 1437-1474.

Zhang (2008b) "On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias," Artificial Intelligence 172: 1873-1896.

Assumptions on the structural equations

In the methods discussed so far, we've allowed that the structural equations are *arbitrary* unknown functions (at least, in principle!):

$$X_i = f_i(\text{Pa}(X_i, \mathcal{G}), \epsilon_i) \quad \forall i \in \{1, ..., p\}$$

However, an alternative approach to structure learning makes explicit assumptions on the structural equations. Such assumptions can imply asymmetries in the observed data, which can be used to tease apart different structures. For example, consider a linear model:

$$X_i = \sum_{X_j \in \text{Pa}(X_i, \mathcal{G})} \beta_j X_j + \epsilon_i \quad \forall i \in \{1, ..., p\}$$

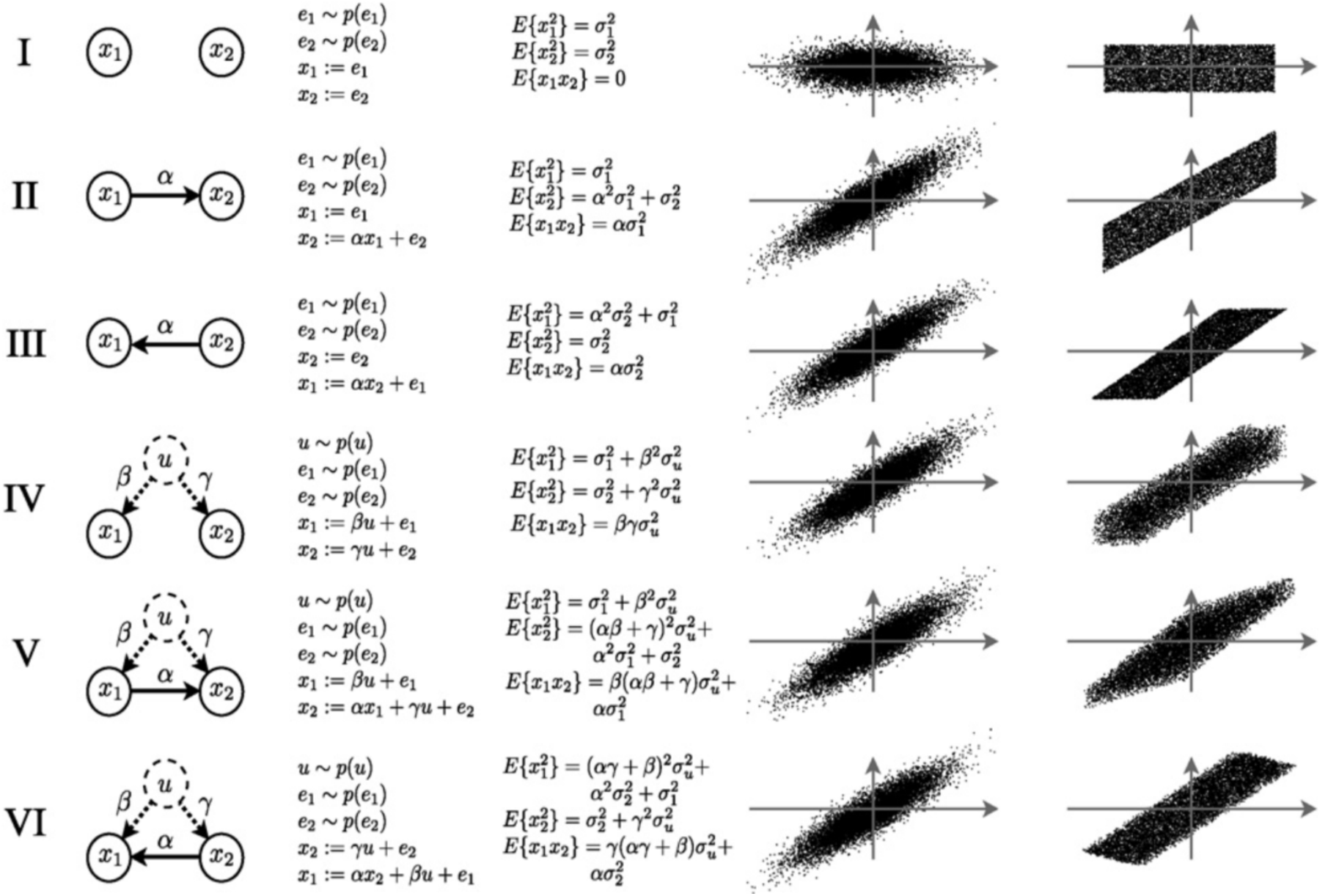
Linear models

Linear models are very common in some areas of applied analysis, particularly because they are convenient to analyze or estimate. However, it is easy to encounter examples for which linearity is obviously false, and **not** an appropriate assumption! If one *does* have reason to expect relationships to be linear, this can be used to significant advantage. Consider the case:

$$X_i = \sum_{X_j \in \text{Pa}(X_i, \mathcal{G})} \beta_j X_j + \epsilon_i$$

$\forall i$ with $\epsilon_1, ..., \epsilon_p$ assumed to be mutually independent and **non-Gaussian**. The combination of linearity and non-Gaussianity[4] makes it possible to identify the direction between variables.

[4]Note: Gaussian error terms + linear functions $\implies$ Gaussian joint distribution. A Gaussian joint distribution $\implies$ linear functions. However, linear functions by themselves do **not** imply Gaussianity: you can have models which are linear, with non-Gaussian errors, which $\implies$ non-Gaussian joint distribution.



From Hoyer et al. (2008) "Estimation of causal effects using linear non-Gaussian causal models with hidden variables," Int. Jour. of Approx. Reasoning 49: 362-378. Last 2 columns show induced distributions over x_1, x_2 with Gaussian and Uniform noise, respectively.

LiNGAM

There are a number of algorithms based on the linear non-Gaussian acyclic model ("LiNGAM"), with or without allowing for latent variables.[5] These typically use results from Independent Component Analysis (ICA) to identify a causal structure consistent with observed data.

When there are no latent variables, these algorithms may identify a unique DAG, rather than an equivalence class. That's because the algorithms exploit information besides conditional independence constraints: the implications of linearity and non-Gaussianity assumptions.
$\Rightarrow$ *you may draw stronger conclusions if you make stronger assumptions; but, those stronger assumptions may be wrong!*

[5] See Shimizu (2014) "LiNGAM: Non-Gaussian methods for estimating causal structures." Behaviormetrika 41(1): 65-98.

LiNGAM

Consider the case with no latent variables, as a matrix equation:

$$X = BX + \epsilon$$
$$X = A\epsilon$$

where $A = (I - B)^{-1}$ and ϵ's are mutually independent. If $B_{ij} \neq 0$ then $X_j \to X_i$. LiNGAM methods use ICA to obtain an estimate of the mixing matrix A.

Actually, ICA typically focuses on estimating the inverse $W = A^{-1}$. Specifically the algorithm will find a matrix $\widehat{W}_*$ such that:

$$\hat{\epsilon} = \widehat{W}_* X$$

with $\hat{\epsilon}$ mutually independent by minimizing $I(\hat{\epsilon}) = \sum_{i=1}^{p} H(\hat{\epsilon}_i) - H(\hat{\epsilon})$ where $H(\hat{\epsilon}) = \mathbb{E}[-\log p(\hat{\epsilon})]$. It can be shown that this mutual information metric is minimized when the elements of ϵ are mutually independent (which is what the model assumes).

LiNGAM

Since ICA only determines $\widehat{W}_*$ up to a permutation of the columns and a scaling factor, the algorithm will permute and normalize the result appropriately to compute $\widehat{B}$, pruning coefficients close to zero if they are "small."

ICA solves the "cocktail party problem": recovering the "source" signals from "microphones" which linearly mix them. Fast algorithms for doing this have been explored in the engineering literature.

To allow for latent variables one may use *overcomplete* ICA: more "sources" than than "microphones."[6]

[6] See Hoyer et al. (2008) "Estimation of causal effects using linear non-Gaussian causal models with hidden variables," Int. Jour. of Approx. Reasoning 49: 362-378.

LiNGAM

Pro:

- By assuming linear non-Gaussian SEMs, algs can sometimes identify a unique DAG (assuming no unmeasured confounding) or a small equivalence class (allowing for unmeasured confounding)

Cons:

- Even with non-Gaussian errors, linearity assumption is very strong and unlikely to hold
- Statistical (asymptotic) properties of ICA-based algorithms are unknown/complicated, may depend on "degree of non-Gaussianity"
- In practice requires large sample sizes and small dimension p to behave well

Additive noise nodels

Another class of models assumes only that the noise terms enter into the function additively:

$$X_i = f_i(\text{Pa}(X_i, \mathcal{G})) + \epsilon_i$$

the functions f_i may be nonlinear (though are usually assumed to be differentiable). Somewhat surprisingly, assuming additive Gaussian noise + nonlinear functions is sufficient to identify the causal structure.

What does this rule out? Multiplicative noise, linear-Gaussian...

Note that the ANM is not closed under marginalization. If you start with an ANM over $V = X \cup L$, then the marginal model over X may no longer be in the ANM class. $\implies$ in settings with latent variables, ANMs are difficult to justify.

Post-nonlinear causal models

Another class of models adds a nonlinear transformation on the additive noise model. Consider $X_j \to X_i$. The PNL model asserts

$$X_j = f_2(f_1(X_i) + \epsilon_j)$$

If the opposite direction $X_j \to X_i$ holds true, then

$$X_i = g_2(g_1(X_j) + \epsilon_i)$$

One may prove that under some technical conditions, $X_j \to X_i$ and $X_j \leftarrow X_i$ can be distinguished from the data.

Exploiting asymmetries

All of these methods impose some assumptions/restrictions on the structural equations, and derive some asymmetry in the observed data distribution from these assumptions. Then, we check if the data exhibits the supposed asymmetry to try and infer backwards to the generating model.

In general, it difficult to establish properties of such methods and also computationally quite difficult to scale them up to large multivariate systems. However, they can sometimes be combined with nonparametric methods like PC, etc to get more informative output. Also, new methods are being developed all the time.

Structure learning algorithms... incomplete list

DAGs/CPDAGs:

- PC (+ variants), GES, ARGES, GSP, MMHC, ICA-LiNGAM, directLiNGAM, CAM, NOTEARS (+ variants)

MAGs/PAGs/ADMGs:

- FCI, RFCI, FCI+, GFCI, GSPo, LV-LiNGAM, M3HC, DCD(+)

Cyclic graphs:

- CCD, LiNG, bcause, Two-Step, SAT-methods

constraint-based, score-based, hybrid, semiparametric, other

Software packages for structure learning

R:
pcalg
bnlearn

Python:
pcalg.py
causal discovery toolbox (cdt)
ananke

Java:
TETRAD

Matlab:
Bayes net toolbox

+ various implementations available from authors of papers