Efficient Adjustment and Causal Discovery

Robin J. Evans and Vanessa Didelez

University of Oxford and BIPS, University of Bremen

APTS Week 4, Glasgow
August 2023

Outline

1. Efficient Adjustment

2. Causal Model Selection
   - Markov equivalence
   - Causal Discovery

Adjustment Sets

We know that, in a directed acyclic graph model, we can estimate the causal effect of (say) A on Y by **adjusting** for a set of variables that forms a **back-door** (adjustment) set.

That is, if we want the causal effect of A on Y, and there is a back-door adjustment set $\boldsymbol{C}$, evaluate

$$\mathbb{E}\,Y(a) := \mathbb{E}_{\boldsymbol{C}}\mathbb{E}[Y \mid A = a, \boldsymbol{C}].$$

What other sets might be valid (i.e. lead to a consistent estimate)?

Which is the most **efficient** set to use?

Here we take 'efficient' to mean having the smallest variance over all possible valid adjustment sets.

Variances

For illustrative purposes, we consider the linear case, but the results extend to general models (Rotnitzky and Smucler, 2020).

Lemma

Suppose we have a multivariate model with covariance matrix $\blacksquare$*. In a linear regression of* Y *on* $\boldsymbol{C}$*, the covariance matrix of the regression coefficient vector is*

$$(n\blacksquare_{\boldsymbol{CC}})^{-1}\sigma_{yy\cdot\boldsymbol{C}}.$$

This leads to a corollary that if we regress Y on A and the collection of variables $\boldsymbol{C}$, the variance of that coefficient is

$$n^{-1}\frac{\sigma_{yy\cdot a\boldsymbol{C}}}{\sigma_{aa\cdot\boldsymbol{C}}}.$$

Independences

Theorem

Suppose that $\boldsymbol{C}, \boldsymbol{D} \subseteq \boldsymbol{V} \setminus \{A, Y\}$*, and let* $\boldsymbol{C}' = \boldsymbol{C} \setminus \boldsymbol{D}$ *and* $\boldsymbol{D}' = \boldsymbol{D} \setminus \boldsymbol{C}$*. Then if* $Y \perp_d \boldsymbol{D}' \mid \boldsymbol{C}, A$ *and* $A \perp_d \boldsymbol{C}' \mid \boldsymbol{D}$*, we have*

$$\frac{\sigma_{yy \cdot a\boldsymbol{C}}}{\sigma_{aa \cdot \boldsymbol{C}}} \leq \frac{\sigma_{yy \cdot a\boldsymbol{D}}}{\sigma_{aa \cdot \boldsymbol{D}}}.$$

In other words, $\boldsymbol{C}$ is a more efficient set to use than $\boldsymbol{D}$!

Proof.

Note that $\boldsymbol{C}' \cup \boldsymbol{D} = \boldsymbol{C} \cup \boldsymbol{D}'$. By the first d-separation, we have

$$\sigma_{yy \cdot a\boldsymbol{C}} = \sigma_{yy \cdot a\boldsymbol{C}\boldsymbol{D}'} = \sigma_{yy \cdot a\boldsymbol{C}'\boldsymbol{D}}.$$

Removing entries from the conditioning set will only increase the residual variance, so $\sigma_{yy \cdot a\boldsymbol{C}} \leq \sigma_{yy \cdot a\boldsymbol{D}}$.

Similarly, using the other independence $\sigma_{aa \cdot \boldsymbol{C}} \geq \sigma_{aa \cdot \boldsymbol{D}}$; hence the result. □

Efficient Adjustment Set

Definition

Let $\text{cn}_{\mathcal{G}}(A \to Y)$ be all nodes on any causal path from A to Y, excluding A.

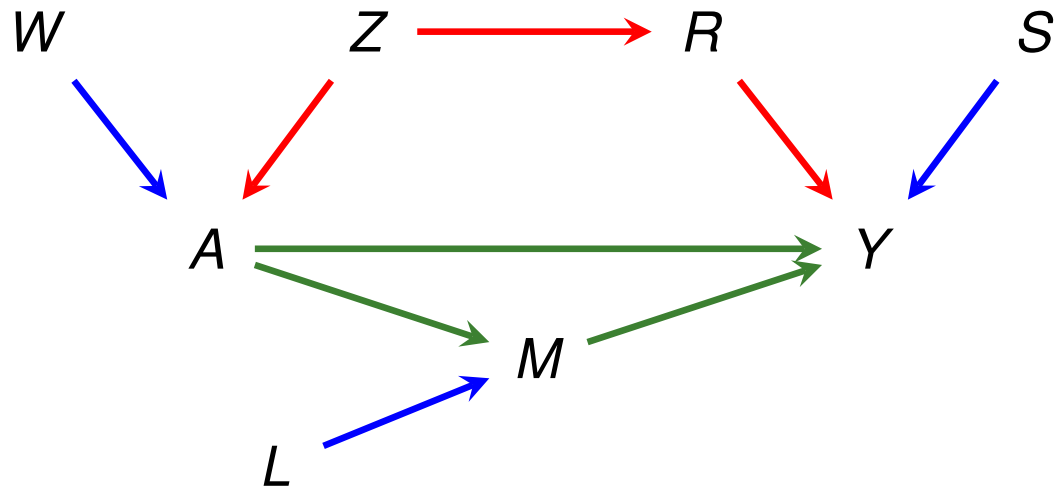
The following result was first proven by Henckel et al. (2022) (preprint 2019), and later extended to the general semi-parametric case by Rotnitzky and Smucler (2020).

Theorem (Efficient adjustment)

$$\textit{Let} \qquad O_{\mathcal{G}}(A \to Y) := \text{pa}_{\mathcal{G}}(\text{cn}_{\mathcal{G}}(A \to Y)) \setminus (\text{cn}_{\mathcal{G}}(A \to Y) \cup \{A\}).$$

Then $O_{\mathcal{G}}(A \to Y)$ is the optimal adjustment set for estimating the causal effect of A on Y, in the sense that the variance of the A-coefficient is minimized.

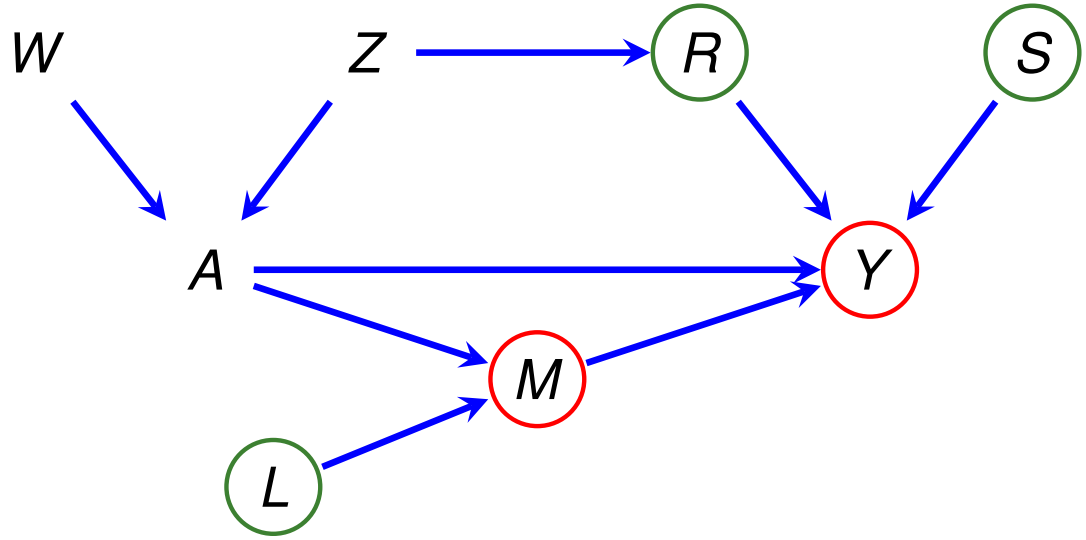
Adjustment Sets



In this graph we:

- must leave causal paths open, so do **not** adjust for M (or A or Y);
- need to block back-door path, so **must** adjust for Z, R or both;
- can *decide whether* to adjust for any of W, L, S.

Efficient Adjustment



We have

$$O_{\mathcal{G}}(A \to Y) := \mathrm{pa}_{\mathcal{G}}(\mathrm{cn}_{\mathcal{G}}(A \to Y)) \setminus (\mathrm{cn}_{\mathcal{G}}(A \to Y) \cup \{A\}),$$

where $\mathrm{cn}_{\mathcal{G}}(A \to Y) = \{M, Y\}$, and therefore the first set is $\{A, S, M, L, R\}$.

Hence $O_{\mathcal{G}}(A \to Y) = \{L, R, S\}$.

Forbidden Projection

Another way to characterize $O_{\mathcal{G}}(A \to Y)$ is via the **forbidden projection**.

The **forbidden set** is $\text{forb}_{\mathcal{G}}(A \to Y) = \text{de}_{\mathcal{G}}(\text{cn}_{\mathcal{G}}(A \to Y)) \cup \{A\}$.

This is the set of nodes that never appear in **any** valid adjustment set.

Forbidden Projection

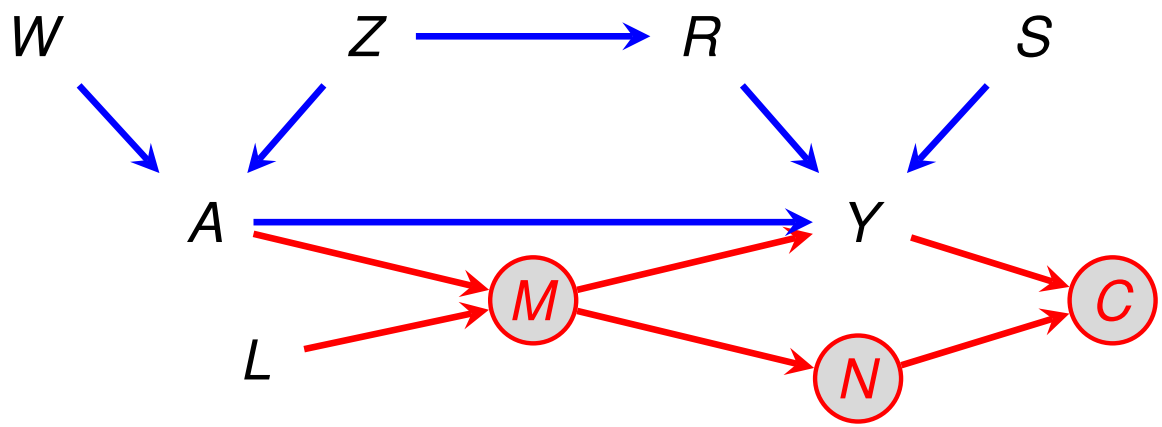
This consists of:

(i) projecting out the forbidden nodes (except A and Y) to obtain $\mathcal{G}'$;

(ii) setting $O_{\mathcal{G}}(A \to Y) = \text{pa}_{\mathcal{G}'}(Y) \setminus \{A\}$.

To perform **latent projection**, any vertices to be dropped that have no children are simply removed. (Why?)

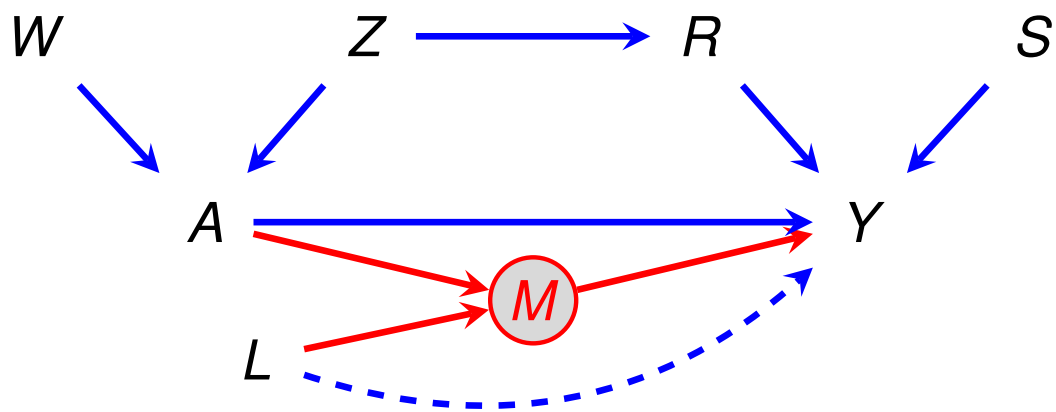
Forbidden Projection



To project out:

- remove vertices with no observed descendants;
- replace any mediators with edges directly from parents to their children.
- then drop the mediators.

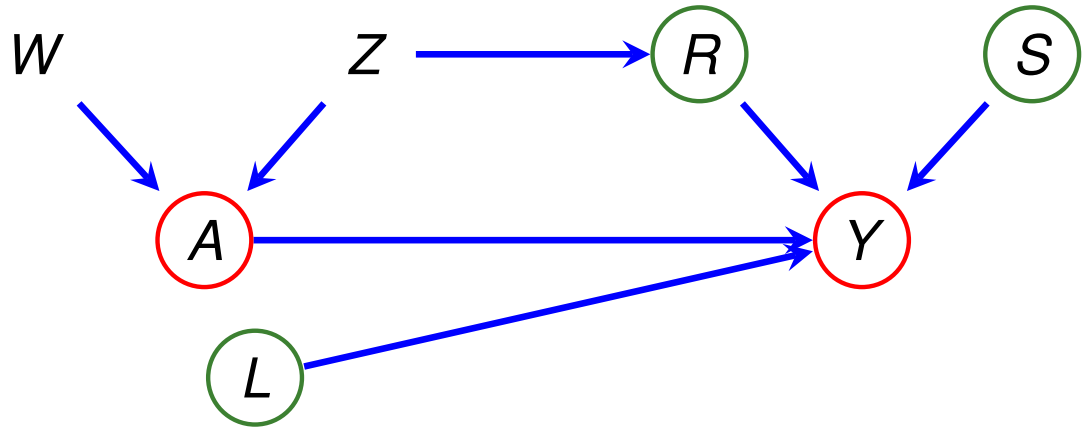
Forbidden Projection



To project out:

- remove vertices with no observed descendants;
- replace any mediators with edges directly from parents to their children.
- then drop the mediators.

Forbidden Projection



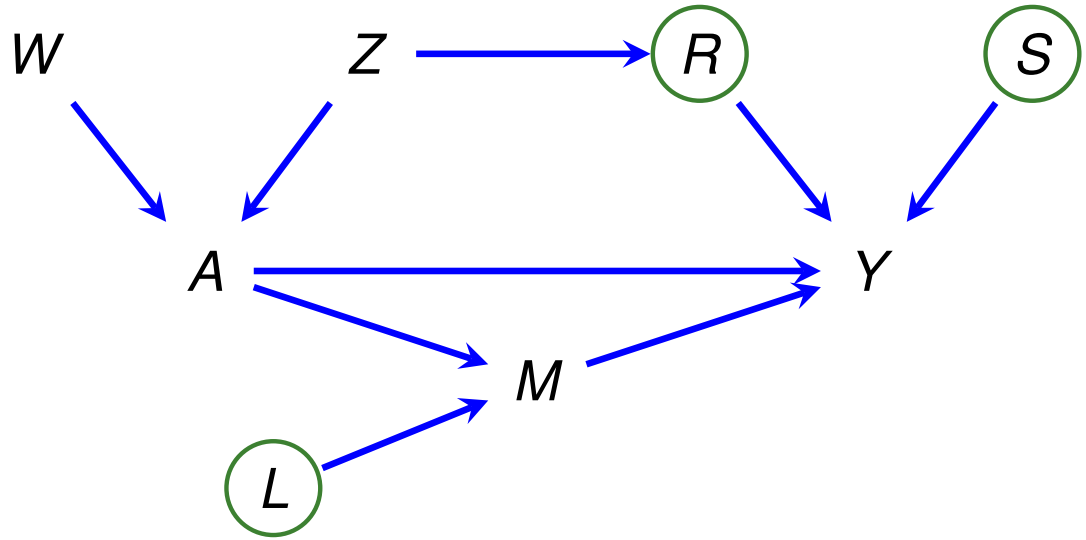
To project out:

- remove vertices with no observed descendants;
- replace any mediators with edges directly from parents to their children.
- then drop the mediators.

Now notice that $O_{\mathcal{G}}(A \to Y) = \mathrm{pa}_{\mathcal{G}'}(Y) \setminus \{A\}$.

Intuition behind Efficient Adjustment

Notice that we adjust for some variables (L and S), even though these are **not** actually confounders.



Notice also that we **do not** control for instruments. (i.e. variables affecting only treatment).

In theory conditioning on an instrument will **increase** the variance in the estimate, because it **reduces** variance in A.

In practice, conditioning on an instrument will also induce **bias**.

Intuition behind Efficient Adjustment

Think of effect estimation as a regression.

```
X <- rnorm(100, sd=1)
Y <- X + rnorm(100, sd=1)
summary(lm(Y ~ X))$coef[,1:2]


##             Estimate Std. Error
## (Intercept)    -0.10       0.098
## X               0.95       0.107


X <- rnorm(100, sd=0.1)
Y <- X + rnorm(100, sd=1)
summary(lm(Y ~ X))$coef[,1:2]


##             Estimate Std. Error
## (Intercept)    -0.03       0.11
## X               0.51       1.10
```

Reducing the variation in X **increases** the standard error.

Intuition behind Efficient Adjustment

Think of effect estimation as a regression.

```
X <- rnorm(100, sd=1)
Y <- X + rnorm(100, sd=1)
summary(lm(Y ~ X))$coef[,1:2]


##             Estimate Std. Error
## (Intercept)    -0.10       0.098
## X               0.95       0.107


X <- rnorm(100, sd=1)
Y <- X + rnorm(100, sd=0.1)
summary(lm(Y ~ X))$coef[,1:2]


##             Estimate Std. Error
## (Intercept)   -0.003       0.011
## X              0.995       0.011
```

However, reducing the variation in Y **decreases** the standard error.

More about Efficient Adjustment

The key quantity is:

$$\frac{\text{variance in } Y}{\text{variance in } X}.$$

We want the top to be small and the bottom to be large for good precision.

The results in the earlier slides were first proved in the multivariate Gaussian case by Henckel et al. (2022) (preprint 2019).

The forbidden projection idea comes from Witte et al. (2020).
Gives a nice duality:

- **worst** adjustment set is parents of treatment;
- **best** adjustment set is parents of outcome (after projection!)

More about Efficient Adjustment

It was extended to the general semi-parametric case by Rotnitzky and Smucler (2020).

In fact, they show that any non-parametric estimation method (i.e. without making use of parameteric assumptions in the conditional distributions) can be performed most efficiently using the optimal adjustment set.
This includes **propensity score** methods, **doubly robust** approaches and **double machine learning** methods.

It has also been generalized to models that have hidden variables, though the results are not always so nice (see Smucler et al., 2020).

References

Henckel et al. – Graphical criteria for efficient average causal treatment effect estimation via adjustment in causal linear models. *JRSS-B*, 2022.

Rotnitzky and Smucler – Efficient adjustment sets for population average causal treatment effect estimation in graphical models. *JMLR*, 2020.

Smucler et al. – Efficient adjustment sets in causal graphical models with hidden variables. *Biometrika*, 2022.

Witte et al. – On Efficient Adjustment in Causal Graphs, *JMLR*, 2020.

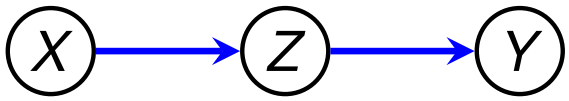
Outline

1. Efficient Adjustment

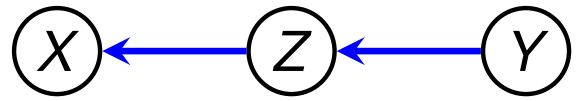
2. Causal Model Selection
   - Markov equivalence
   - Causal Discovery

Markov equivalence

Notice that the Markov properties of **distinct** graphs are sometimes the same.



$$p(x) \cdot p(z \mid x) \cdot p(y \mid z)$$



$$p(y) \cdot p(z \mid y) \cdot p(x \mid z)$$

$$p(z) \cdot p(x \mid z) \cdot p(y \mid z)$$

All these graphs imply (precisely) that $X \perp\!\!\!\perp Y \mid Z$.

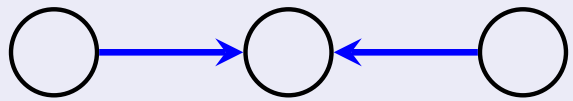
If two different graphs define the same model, we say that they are **Markov equivalent**.

A set of graphs that are all Markov equivalent is called the **Markov equivalence class**.

PDAGs and CPDAGs

This means that, given data, we cannot tell what the precise underlying DAG that generated the data was!

The Markov equivalence class of a DAG is determined by its skeleton (i.e. which nodes are adjacent) and its **unshielded colliders** or **v-structures**; that is induced subgraphs isomorphic to:



Instead we can represent the **pattern**, by orienting any triples with precisely two edges if they are a v-structure.

If the graph is compatible with a DAG we call it a **partially directed acyclic graph** (PDAG).

Other orientations follow logically by Meek's rules (see Appendix). Applying exhaustively means we obtain a **completed** PDAG (CPDAG).

What can we do if we don't know the model?

Typically, we assume that there **is** some DAG model that explains the data we observe.

There are then various approaches we can use:

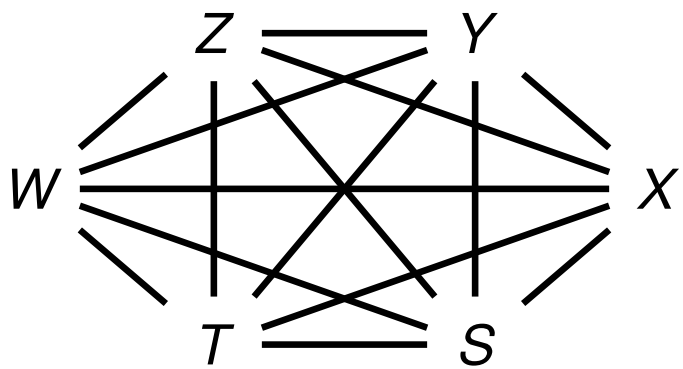
- learn the DAG from observational data alone
  (e.g. PC/FCI algorithm, GES, order/partition MCMC, SAT solvers);
- if available, use data from interventions or perturbations of the model (e.g. invariant prediction, GIES);
- use non-linearities or non-Gaussian errors to exploit asymmetries
  (e.g. LiNGAM, causal additive noise models).

Constraint-Based Approaches

These use signficance tests to decide whether constraints hold or not.

The **PC algorithm** (Spirtes et al. 2000) works by starting with a complete graph, and testing all marginal independences.

Any hypothesis **not** rejected (at size α) leads to the removal of an edge, and the empty conditioning set is recorded.



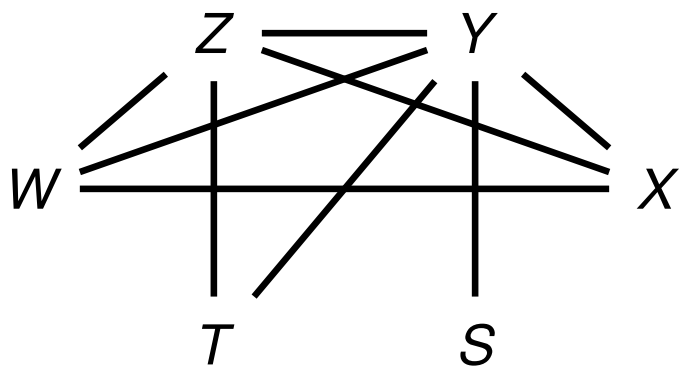
The algorithm then proceeds to conditioning sets of size 1, 2, etc., searching only through the neighbours of one of the two vertices.

Constraint-Based Approaches

These use signficance tests to decide whether constraints hold or not.

The **PC algorithm** (Spirtes et al. 2000) works by starting with a complete graph, and testing all marginal independences.

Any hypothesis **not** rejected (at size α) leads to the removal of an edge, and the empty conditioning set is recorded.



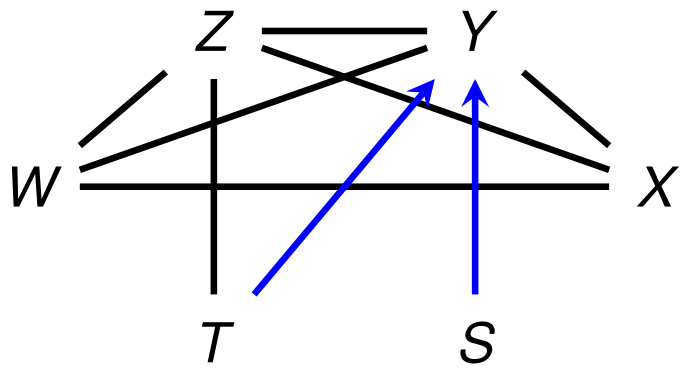
The algorithm then proceeds to conditioning sets of size 1, 2, etc., searching only through the neighbours of one of the two vertices.

Constraint-Based Approaches

These use signficance tests to decide whether constraints hold or not.

The **PC algorithm** (Spirtes et al. 2000) works by starting with a complete graph, and testing all marginal independences.

Any hypothesis **not** rejected (at size α) leads to the removal of an edge, and the empty conditioning set is recorded.



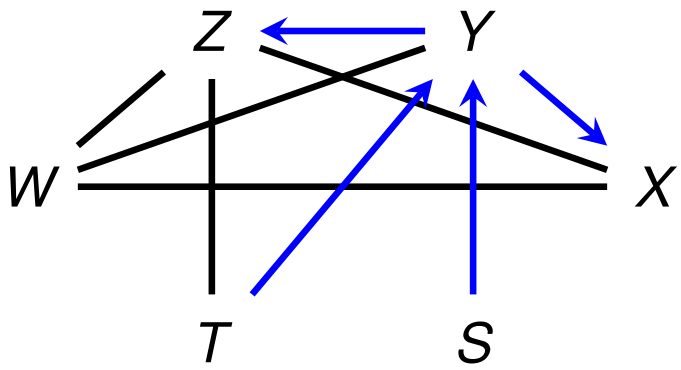
After determining the skeleton, it goes back to see how unshielded triples should be oriented based on the set that made the variables conditionally independent.

Constraint-Based Approaches

These use signficance tests to decide whether constraints hold or not.

The **PC algorithm** (Spirtes et al. 2000) works by starting with a complete graph, and testing all marginal independences.

Any hypothesis **not** rejected (at size α) leads to the removal of an edge, and the empty conditioning set is recorded.



After determining the skeleton, it goes back to see how unshielded triples should be oriented based on the set that made the variables conditionally independent.

It then uses Meek's rules to obtain the CPDAG.

Constraint-Based Approaches

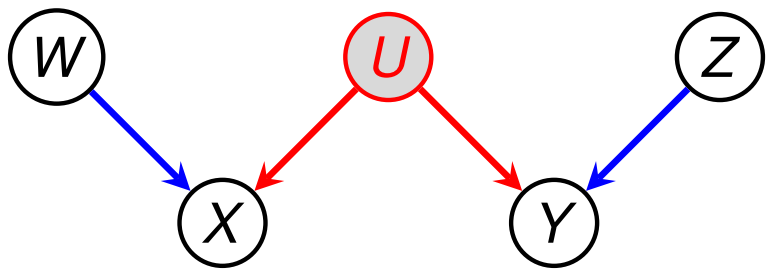
This algorithm is **very efficient** if the graph is sparse.

THe PC algorithm requires an assumption of **faithfulness** to work. That is:

$$X \perp\!\!\!\perp Y \mid Z\,[p] \qquad \Rightarrow \qquad X \perp_d Y \mid Z\,[\mathcal{G}].$$

This is the converse of p being **Markov** w.r.t. $\mathcal{G}$.

Since $X \perp\!\!\!\perp Y$ and $X \perp\!\!\!\perp Y \mid Z$ can look very similar (Evans, 2020), it will typically remove edges too freely, and it is **not guaranteed** to return a CPDAG!



Alternatives (e.g. conservative PC) also exist.

Score-Based Approaches

Score-based approaches attempt to maximize a score that will be consistent for the true model.

Example

The classic score-based approach for DAGs is **greedy equivalence search** (GES).

This has two phases; starting with an empty graph:

1. add the edge to the PDAG that maximizes the local increase in the score, stopping when no improvement is possible;
2. remove the edge that maximizes the local increase in the score, stopping when no improvement is possible.

For sufficiently large samples, GES is guaranteed to find the Markov equivalence class of the true DAG (Chickering, 2002).

Hybrid Approaches

PC and GES both have high-dimensional guarantees (e.g. Kalisch and Bühlmann, 2007, Nandy et al., 2018).

However, we generally have the following trade-off to make:

- constraint-based approaches are fast, but often make mistakes that cascade into further errors;
- score-based approaches are more accurate, but take longer to run.

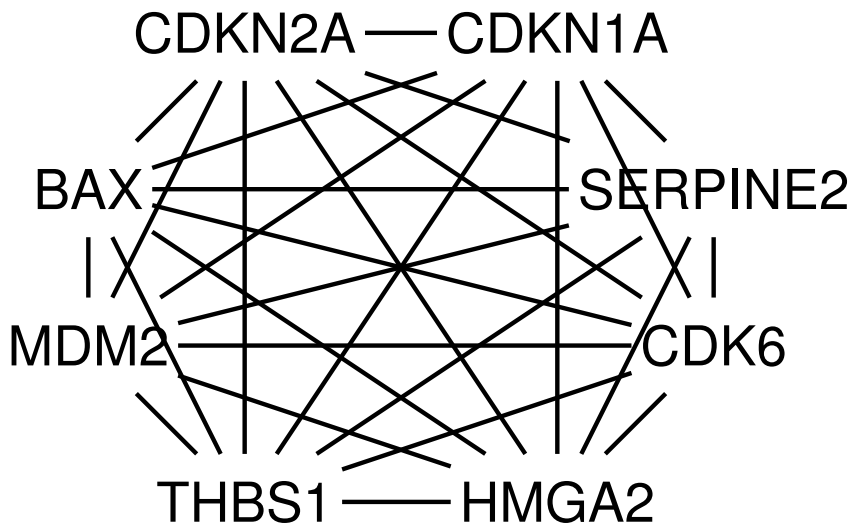
This motivates **hybrid approaches**, which typically start by reducing the search space using constraints, and then proceed to use a more accurate score-based method on the smaller set of models.

Examples include Max-Min Hill Climbing, or running the first phase of PC and then GES on the resulting skeleton.

See Scutari and Denis (2021) for examples in R.

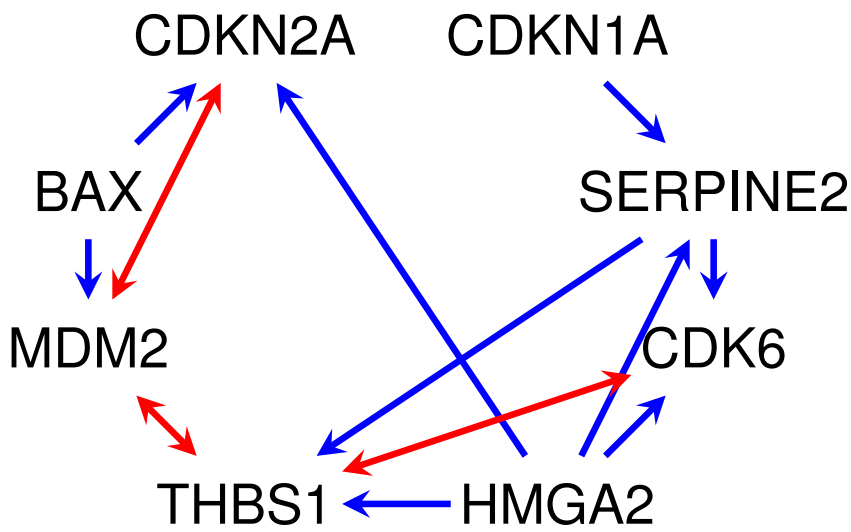
Causal Search Example

The dataset `tcgas` contains expression data from eight genes in tumour tissues taken from $n = 362$ patients with squamous cancer of the head and neck (Foraita et al., 2020).



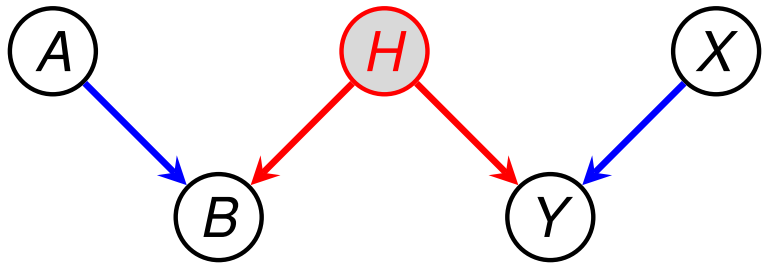
Causal Search Example

The dataset `tcgas` contains expression data from eight genes in tumour tissues taken from $n = 362$ patients with squamous cancer of the head and neck (Foraita et al., 2020).



Hidden Variables

There are configurations of independences from DAGs that give these **bidirected** edges, but they require **unobserved variables**.



This extends the ‘projection’ operation given in the efficient adjustment section.

The result on the previous slide suggests that hidden variables should be allowed for, which would mean using the **fast causal inference** (FCI) algorithm instead (Spirtes et al., 2000).

LiNGAM

All the approaches so far can only discover the **Markov equivalence class** of the model.

Other assumptions and new methods are required to determine causal directions within these classes.

One such approach is **Linear, Non-Gaussian Acyclic Models** (LiNGAM), which assumes that the noise is **not** Gaussian.

The idea is then very similar to **independent components analysis** (ICA), where the sum of non-Gaussian variables can be disentangled.

We then have, for $X \in \mathbb{R}^p$,

$$X = BX + \varepsilon \quad \Longrightarrow \quad X = (I - B)^{-1}\varepsilon.$$

ICA aims to deduce $W = I - B$, and hence obtain the original sources $\varepsilon \in \mathbb{R}^p$.

LiNGAM Algorithm

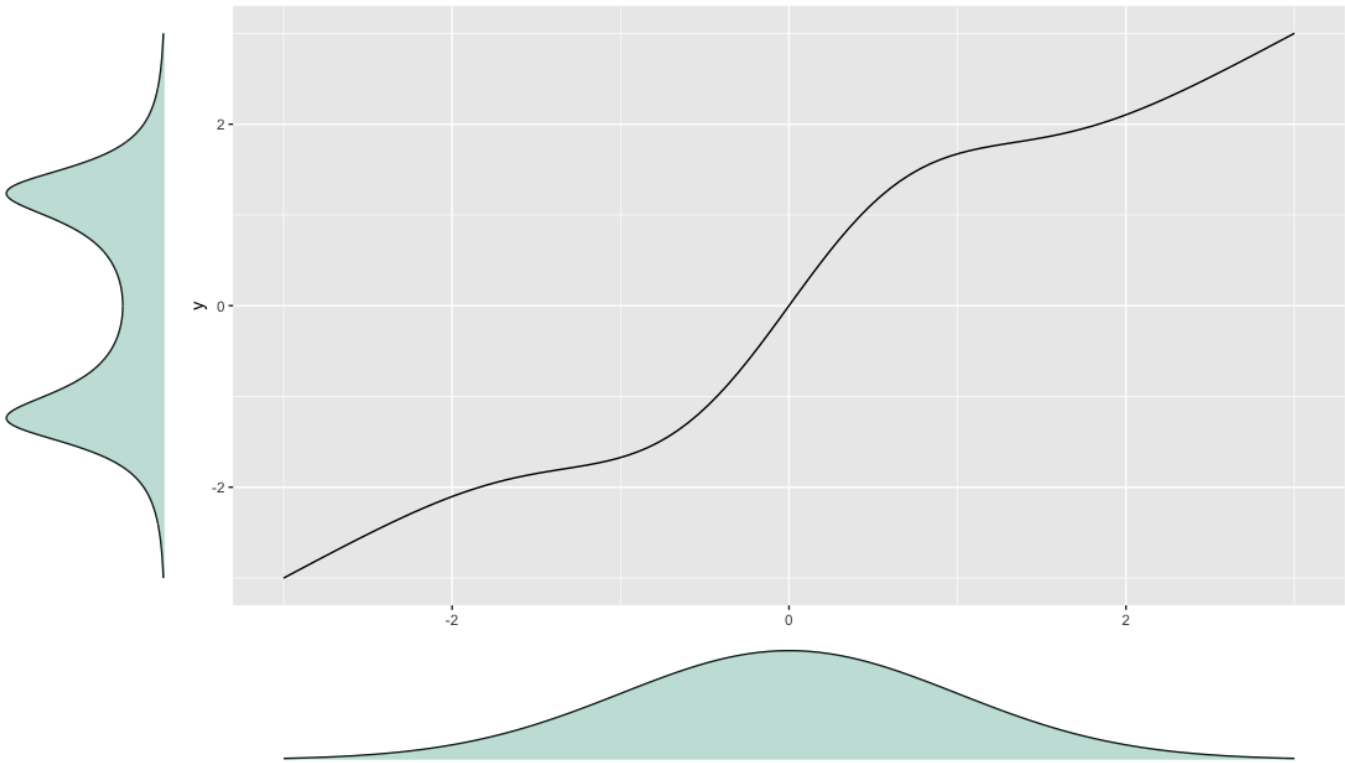
In ICA, we are given n samples of a p-dimensional vector and asked to 'disentangle' the sources.

It is crucial that none (or at most one) of the sources is Gaussian, otherwise the original sources become unidentifiable.

1. Given an $n \times p$ data matrix X ($n \gg p$), centre each column.
2. Apply the ICA algorithm $X = SW^{-1}$ where $S \in \mathbb{R}^{n \times p}$ contains independent components in its rows.
3. Find unique permutation of columns of W s.t. no zeros on diagonal.
4. Divide each column of W by its corresponding diagonal element.
5. Compute an estimate $\hat{B} = I - \tilde{W}$.
6. Get permutation of $\hat{B}$ that is lower triangular.

Information Geometric Approach

For this approach we need a **postulate** (assumption): that the **mechanisms** that generate values of the **cause** should be unrelated to the mechanism that turns causes into **effects** (Daniusis, 2010).

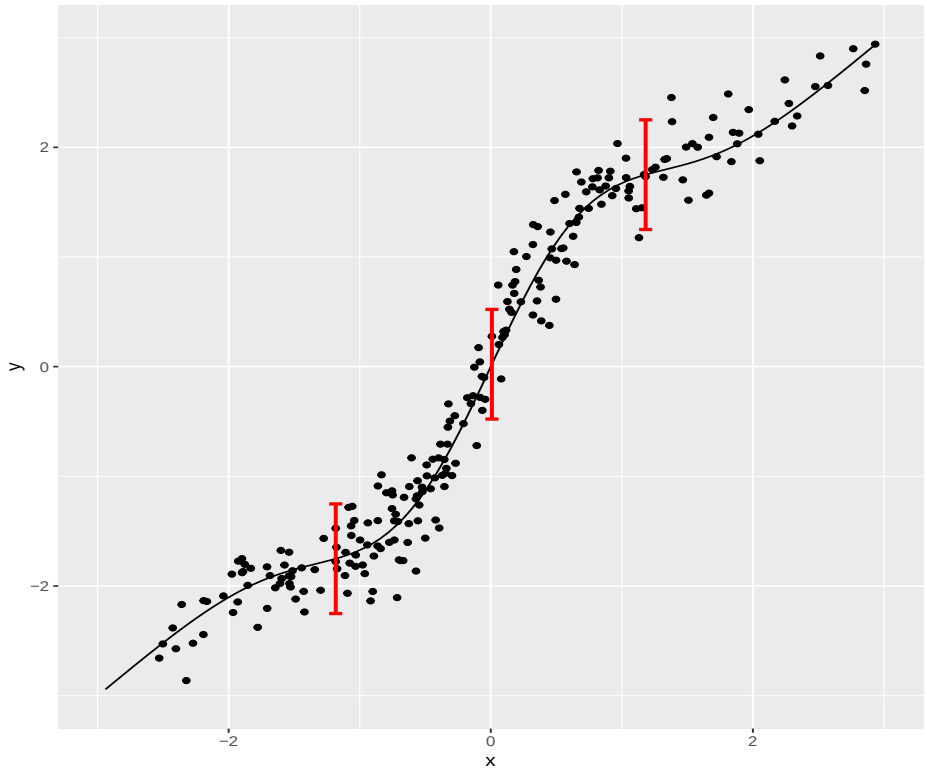


It will then follow that the distribution of the cause is unrelated to the mechanism, while the effect **is** 'correlated' with it.

Additive Noise Models

Another approach is to assume an **additive noise model**:

$$Y = f(X) + \epsilon, \qquad \epsilon \perp\!\!\!\perp X.$$

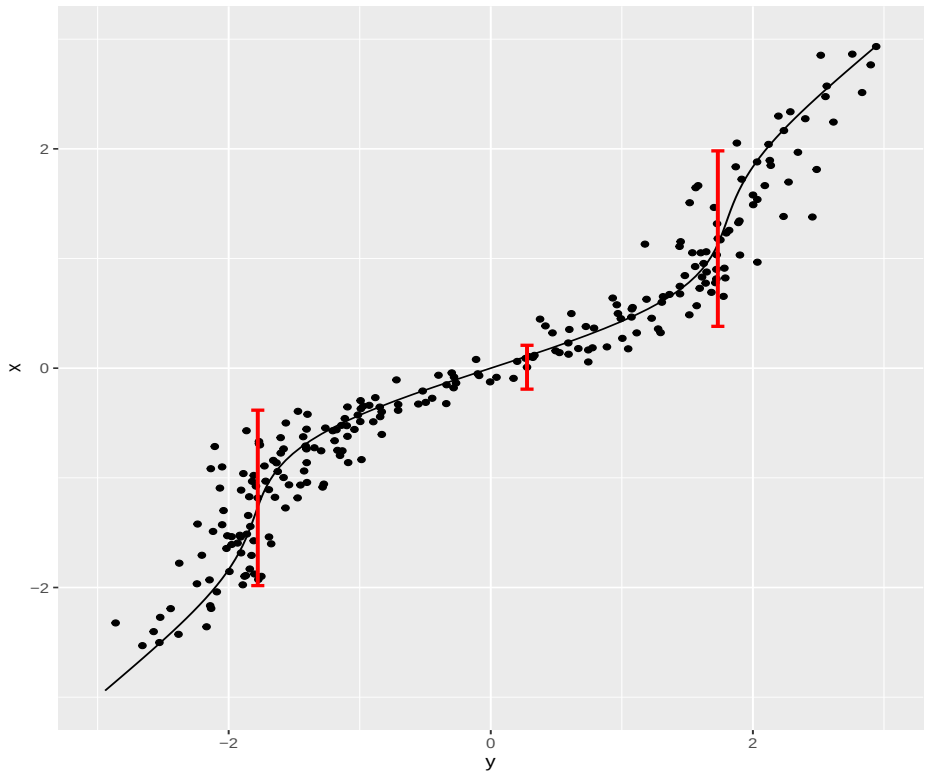


This way around the noise distribution looks similar at all points.

Additive Noise Models

However, if we flip the picture...

$$X \neq g(Y) + \delta, \qquad \delta \perp\!\!\!\perp Y.$$



...now clear dependency between steepness of the curve and variance.

RESIT

This motivates the Regression and Subsequent Independent Test (RESIT) algorithm of Peters et al. (2014).

First, perform a (flexible) regression of Y on X to give residuals r_Y and fitted values f_X.

Then do the same for X on Y to give residuals r_X and fitted values f_Y.

Then see whether $r_Y \perp\!\!\!\perp f_X$ or $r_X \perp\!\!\!\perp f_Y$ is more plausible, and determine the causal direction accordingly.

Note that, for this method to work, we **cannot** have a linear function $f : X \mapsto Y$ **and** a Gaussian noise distribution (the only symmetric case!)

NOTEARS

The optimization procedure we have described is effectively **discrete**, because we have to search over a space of graphs.

Several recent methods (Zheng et al., 2018) instead use a **continuous function** to enforce acyclicity. For example:

$$\text{tr}(\exp\{A \circ A\}) - d = 0 \quad \iff \quad \mathcal{G} \text{ is acyclic,}$$

where A is the $d \times d$-adjaceny matrix for $\mathcal{G}$, and $\circ$ represents pointwise multiplication.

Combined with an assumption of additive noise models makes NO TEARS a state-of-the-art approach.

Other approaches **combine** a structure learning method (such as NO TEARS) with methods for estimating causal effects. See, for example, Geffner et al. (2022) or Sharma and Kiciman (2020).

Other Methods

There are **countless** other methods that exist (Vowels et al., 2022):

Method	Year	Type	Suff.	Faith.	Acycl.	Interv.	Output
PC [228]	1993	constraint	yes	yes	yes	no	CPDAG
FCI [2							
CCD [							
TPDA							
CPC [							
KCL [2							
ION [2							
IDA [1							
GSF [103]							
bQCD [235]							
GCL [251]							
GGIM [55]							
GYKZ [68]							
SLARAC etc. [260]							
Order-MCMC [60]							
OG [53]							
EE-DAG [281]							
ZIPBN [35]							
LiNGAM [221]							
LV LiNGAM [101]							
non-linear ANM [							
PNL [277]							
CAN [113]							
CCM [232]							
IGCI [112]							
KCDC [161]							
MMHC [245]							
ARGES [172]							

Method	Year	Data	Form	Acycl.	Interv.	Output
EG [52]						
TWILP [18						
CAM [26]						
K2 [38]						
LB-MDL [						
HGC [91]						
CMS [152]	2014	low, dynamic/time series	NN	–	no	direction
NO TEARS [279]	2018	low	linear	yes	no	DAG
CGNN [74]	2018	low	NN	yes	no	DAG
SAM [121]	2019	low/medium	NN	yes	no	DAG
DAG-GNN [272]	2019	low	NN	yes	no	DAG
GAE [178]	2019	low	NN	yes	no	DAG
NO BEARS [142]	2019	low/medium/high	3-poly	yes	no	DAG
DEAR [219]	2020	image	NN	yes	no	–
CAN [167]	2020	low/medium/image	NN	yes	no	DAG
NO FEARS [259]	2020	low	linear	yes	no	DAG
GOLEM [177]	2020	low	linear	yes	no	DAG
ABIC [19]	2020	low	linear	yes	no	ADMG/PAG
DYNOTEARS [179]	2020	low	linear	yes	no	SVAR
SDI [124]	2020	low	NN	yes	yes	DAG
AEQ [63]	2020	Bi	NN	–	no	direction
RL-BIC [284]	2020	low	NN	yes	no	DAG
CRN [125]	2020	low	NN	yes	yes	DAG
ACD [151]	2020	low, time series	NN	Granger	no	time-series DAG
CASTLE (reg.) [138]	2020	low/medium	NN	yes	no	DAG
GranDAG [139]	2020	low	NN	yes	no	DAG
MaskedNN [176]	2020	low	NN	yes	no	DAG
CausalVAE [267]	2020	image	NN	yes	yes	DAG
CAREFL [126]	2020	low	NN	yes	no	DAG / direction
Varando [251]	2020	low	linear	yes	no	DAG
NO TEARS+ [280]	2020	low	non-linear	yes	no	DAG
ICL [258]	2020	low	NN	yes	no	DAG
LEAST [283]	2020	low/medium/high	linear	yes	no	DAG
CausalMosaic [263]	2020	Bi	NN	–	no	direction
NSM [253]	2021	video, dynamic/time series	NN	–	no	direction

References

Chickering, D. M. Optimal structure identification with greedy search. Journal of machine learning research, 3(Nov), 507-554, 2002.

Evans, R.J. Model selection and local geometry. *Annals of Statistics*, 2020.

Foraita, R. et al. Causal discovery of gene regulation with incomplete data. *JRSS-A*, 183(4), pp.1747-1775, 2020.

Geffner, T. et al. Deep end-to-end causal inference. *arXiv:2202.02195*, 2022.

Kalisch, M. and Bühlman, P. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8(3), 2007.

Nandy, P., Hauser, A. and Maathuis, M. H. High-dimensional consistency in score-based and hybrid structure learning. *Annals of Statistics*, 46(6A), 3151-3183, 2018.

References (cont.)

Peters, J., Mooij, J.M., Janzing, D. and Schölkopf, B. Causal discovery with continuous additive noise models. 2014.

Sharma, A. and Kiciman, E. DoWhy: An end-to-end library for causal inference. *arXiv preprint arXiv:2011.04216*. 2020.

Spirtes, P., Glymour, C.N. and Scheines, R.. *Causation, Prediction, and Search*. MIT press (2nd edition). 2000

Vowels, M. J., Camgoz, N. C., and Bowden, R. D'ya like dags? a survey on structure learning and causal discovery. *ACM Computing Surveys*, 55(4), 1-36, 2022.

Zheng, X., Aragam, B., Ravikumar, P.K. and Xing, E.P. DAGSs with NO TEARS: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31. 2018.