Learning Causal Models

Source: (a) Elements of Causal Inference, Peters et al., Chap 7.
(b) Causality, J. Pearl, 2009. → (aka "text")

Outline:

A. Conditions for learning from (infinite) data
B. Algorithms for structure learning
  └ PC Algorithm for CPDAGs
  └ ICA Algorithm for LINGAMs.

(A.) Sufficient Conditions for Learning with Infinite Samples

(1.) Suppose we are given a DAG G and a joint dist. $p(\cdot)$ over $(X_1, \cdots, X_d)$ that is consistent with G.

Prop: $\exists$ SCM $\mathfrak{C}$ that results in $p(\cdot)$ with Markov factorization given by G.

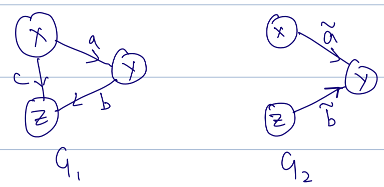
Proof: Straightforward. Iteratively construct the SCM on the DAG starting from the root node.
(See Prop 7.1 in text for details) $\boxtimes$

2. Given infinite samples from $p(\cdot)$, can we recover G?

Not always: Our favorite (counter)example:

Notes 2, pp.24

Recall our discussion on faithfulness — even with infinite samples, we cannot distinguish between the two graphs displayed here.



$X = N_1$
$Y = aX + N_2$
$Z = cX + bY + N_3$.

$X = N_1$
$Y = \tilde{a}X + \tilde{b}X + N_2$
$Z = N_3$

N_1, N_2, N_3 indep., $N_i \sim N(0,1)$.

Suppose $c + ab = 0$. Then, the two paths to Z in G_1 "cancel" each other out, meaning that $X \perp\!\!\!\perp Z$. However $X \not\perp\!\!\!\perp_G Z$.

Faithfulness violation causes us to be unable to distinguish (using even infinite number of samples drawn from $p(\cdot)$) between G_1 and G_2.

3. What if we now impose faithfulness, i.e., we are given infinite samples from $p(\cdot)$ over $(X_1, \ldots, X_d)$ and are told that $p(\cdot)$ is generated by an SCM that is both Markovian (i.e., results in a DAG G^*) and faithful.

Thm (Lemma 7.2 in text): $\nexists\; G' \notin \mathrm{CPDAG}(G^*)$ s.t. $p(\cdot)$ is Markovian and faithful w.r.t. G'.

(Proof immediate from Defn in insert below)

Pasted from Notes 2, pp 20.

Markov Equivalence of DAGs.

Given a directed graph $G = (V, E)$, let $\mathcal{M}(G)$ be the set of all distributions $p(\cdot)$ that have the Markov property w.r.t G, i.e.,

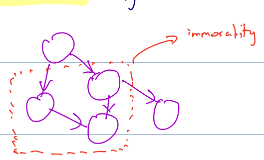
$$\mathcal{M}(G) = \{ p : p(x_1, \dots, x_d) \text{ has the Global Markov property w.r.t } G \}$$

Definition: (Markov Equivalence of Graphs). DAGs G_1 and G_2 are Markov Equivalent if

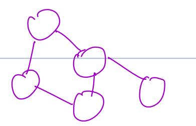
$$\mathcal{M}(G_1) = \mathcal{M}(G_2).$$

In other words, the graphs G_1 and G_2 have the same d-separation relationships, and thus, the same factorization structure and CI relationships among nodes.

Defn: (Skeleton of DAG) The skeleton of a DAG G consists of the vertices along with the undirected edges.



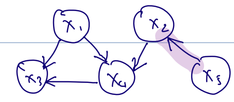
DAG G



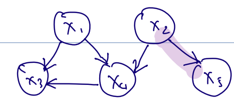
Skeleton of G.

Defn: (Immorality) A collection of three nodes (X, Y, Z) form an immorality if $X \to Y \leftarrow Z$ (i.e., X and Z are parents of Y), but there is no edge between X and Z. (This is also called an unshielded collider)

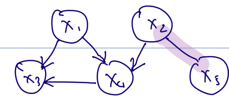
Example (Figure 6.4 in 'Elements of Causal Inference' book, pp. 103)



DAG G_1



DAG G_2



$\text{CPDAG}(G_1) = \text{CPDAG}(G_2)$

Graphs G_1 and G_2 above are Markov Equivalent.

Defn: (Markov Equivalence Class) The set of all DAGs that are Markov Equivalent to G is called its Markov Equivalence Class.

Completed Partially Directed Acyclic Graph

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

$$\text{CPDAG}(G) = \{ (V, E') : \text{directed edge } e \in E' \text{ iff all members of the Markov Equivalence of } G \text{ have the same directed edge; all other edges } e \in E \text{ are represented by undirected edges} \}$$

Lemma 4: G_1 and G_2 are Markov Equivalent

$\Updownarrow$

The graphs have the same skeleton and same imoralities.

Summary: Markov + Faithful $\Longrightarrow$ We can learn the CPDAG if we have access to infinite samples

(B.) Algorithms for Learning DAGs.

In addition to refs so far, see also:

Causal Structure Learning, C. Heinze-Deml, M. Maathuis and N. Meinshausen, arXiv: 1706.09141

https://arxiv.org/abs/1706.09141

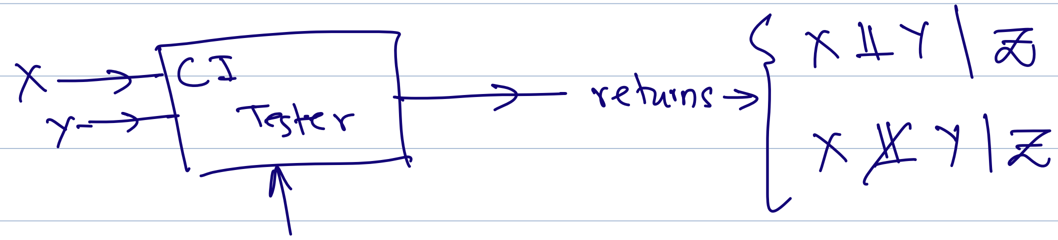
Review of Causal Discovery Methods Based on Graphical Models, Clark Glymour, Kun Zhang and Peter Spirtes, Frontiers in Genetics, June 2019.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6558187/

(1.) PC Algorithm. (Peter Spirtes, Clark Glymour)

Ref: Causation, Prediction and Search, P. Spirtes, C. Glymour and R. Scheines, MIT Press, 2001.

This requires access to a CI Testing Algorithm, i.e.,



$Z \rightarrow \subseteq V \setminus \{X, Y\}$.

(Also, we assume access to a "noiseless" CI Tester, meaning that everything below is in the infinite sample limit. In practice, these algorithms are noisy as we work with only a finite number of samples. There are many other issues that come up then. We will discuss some of them later.

For some refs on non-parametric CI testing, see:

Kernel-based conditional independence test and application in causal recovery, K. Zhang, J. Peters, D. Janzing and B. Scholkopf, UAI 2011. (arXiv 1202.3775)

(Also see Note 8. focussing on CI Testing).

(Back to PC Algorithm).

(a) Learn skeleton, i.e., the undirected graph, using CI Tester.

(b) Orient edges (up to CDPAG).

(a) Learning skeleton.

Key properties (Lemma 7.8 in text)

(i) $X - Y$ are adjacent $\Longleftrightarrow$ they cannot be d-separated by any subset $Z \subseteq V \setminus \{X, Y\}$

(ii) X and Y are not adjacent $\Longrightarrow$ they are d-separated by either PA_X and/or PA_Y.

$\therefore$ Recipe for finding graph using (i): Pick any (X, Y). Search over all $Z \subseteq V \setminus \{X, Y\}$ using CI Tester, i.e, exaustively search if

$$X \perp\!\!\!\perp Y \mid Z \quad \text{for some } Z$$

If no, then $X - Y$. (This was the basic algo, improved by PC Algorithm)

PC Algorithm provides a structured way of executing this search, by using property (ii) above. If we can find a set Z s.t. $X \perp\!\!\!\perp Y \mid Z$, then $X \not- Y$.
no edge

— Start with complete (undirected) graph over all variables.

Set $k = |Z|$.

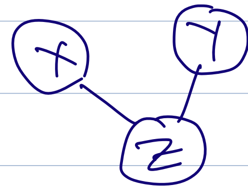
— ($k=0$) For each pair of nodes, check if $X \perp\!\!\!\perp Y$. If you find any, then delete the edge.

— ($k=1$) For each triplet of nodes X, Y, Z, check if $X \perp\!\!\!\perp Y \mid Z$. If so, delete edge between $X - Y$.

— ($k=2$) For each 4-tuple of nodes $X, Y, (Z_1, Z_2)$, ...

⋮

(b) Orienting edges: Set of orientation rules known that are known to be complete (Meek's rules).

e.g.: Suppose we have 

, where $X \not\perp\!\!\!\perp Y$

from part (a). Let Z be a set of nodes that d-separates X and Y. (recall d-separation $\equiv$ CI because of faithfulness assumption).

Then $Z \notin \mathcal{Z} \iff X \rightarrow Z \leftarrow Y$

Summary: Using d-separation $\equiv$ CI, and calls to a CI Tester, we can learn the structure in a population (i.e., infinite sample) setting.

LINGAMs and ICA Algorithm.

Instead of allowing any CPDAG, we can impose restrictions on the SCM, which can allow structure learning. We discuss one approach below using linear non-Gaussian models

LiNGAM – Linear NonGaussian Additive Model

Ref: A linear non-Gaussian acyclic model for causal discovery, S. Shimizu, P. Hoyer, A. Hyvarinen, A. Kerminen, JMLR 2006.

SCM $\mathfrak{C}$: $X_j = \sum_{k \in PA_j} a_{jk} X_k + N_j \, , \; j = 1, 2, \ldots, d.$

where $\{N_j\}$ are mutually independent, and not

Gaussian and not-degenerate (e.g. strictly positive density), and $\mathrm{Var}(N_j) = 1$.

(Aside: What does "not Gaussian" mean: the key property will be that the joint pdf is not rotationally symmetric)

In this case, we can learn the structure using ICA.

$$X_j = \sum_{k \in PA_j} a_{jk} X_k + N_j, \quad j = 1, 2, \ldots, d.$$

i.e., $X = \begin{pmatrix} X_1 \\ \vdots \\ X_d \end{pmatrix}$ $\quad N = \begin{pmatrix} N_1 \\ \vdots \\ N_d \end{pmatrix}$: Then,

$$X = AX + N$$

Since the model is a DAG, we can always reindex variables s.t. A is lower triangular.

$$\therefore \quad X = (I - A)^{-1} N. \doteq BN$$

The problem becomes: We are given samples of X, and we need to learn B. Note that we do not have access to samples of N that generate X; we only know that N is indep. across components, with unit variance and is non-Gaussian.

We now use ICA to determine B, a $d \times d$ matrix. Summary of ICA below:

Let $B = U \Lambda V^T$. Then, $X = U \Lambda V^T N$

We need to learn U, V unitary matrices, and Λ a diagonal matrix.

$$X X^T = U \Lambda V^T N N^T V \Lambda U^T$$

(see below)

$$\therefore E[XX^T] = U \Lambda V^T \underbrace{E[NN^T]}_{\to I} V \Lambda U^T$$

Recall $Var(N_j) = 1$, $\{N_j\}$ indep. Further, WLOG, assume $E[N] = 0$ (else we can work with $\tilde{X} = X - E[X]$, where $E[X]$ can be computed from observed data).

$$\therefore E[XX^T] = U \Lambda \underbrace{V^T V}_{I} \Lambda U^T = U \Lambda^2 U^T \to 1$$

square, symmetric, p.d. covariance matrix

Use PCA to determine Λ^2, U.

$\therefore$ We know $|\lambda_j|$, $j = 1, 2, \ldots, d$ and U a unitary matrix.

Now, $X = BN = (U\Lambda)(V^T N)$

$$Y = \tilde{\Lambda}^{-1} U^T X = \tilde{I} V^T N = \tilde{V}^T N$$

($|\lambda_i|^{-1}$ on diagonal)

± 1 on diagonal, zero elsewhere

Since N is non-Gaussian, zero mean, unit variance noise, its joint pdf is NOT circularly symmetric. $\tilde{V}$ is unitary matrix that rotates N. Let R be any rotation (unitary matrix).

Now, we have access to Y. Search over all RY s.t. $Z = RY$ is independent across coordinates. Then, non-Gaussianity $\Rightarrow R = \tilde{V}$.

Note: The search over rotations and the associated testing for independence across components has several heuristics in literature. Please check out any tutorial / wikipedia for details.