Causal Data Science with Directed Acyclic Graphs (DAGs)

Section 3: Causal Discovery

Dr. Paul Hünermund

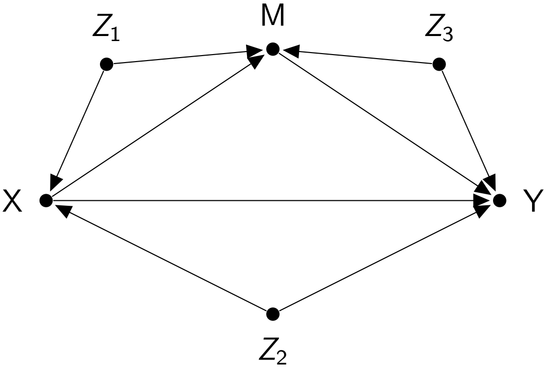
Online Course at Udemy.com

Course Outline

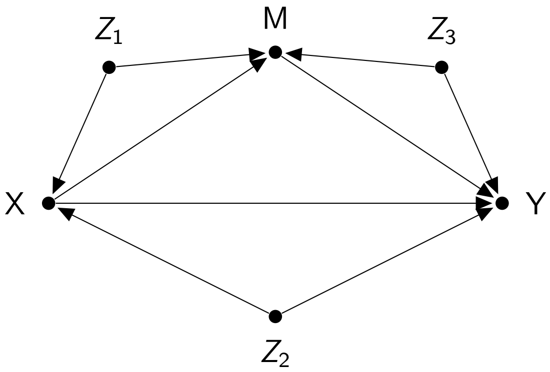
Where do DAGs come from?

- DAGs are a model of how we think the world works
- We arrive at such a model by using our expert knowledge about the particular context under study
    - E.g., by consulting the relevant scientific literature for a topic
- One big advantage of DAGs is that they give rise to testable implications based on the d-separation relations implied by the model (Pearl et al., 2016)
    - If the data are not compatible with the implied d-separation relations, we can discard the graph and build a new one
    - Compared to global "goodness of fit" measures used in the traditional SEM literature, these testable implications are local and provide the analyst with more fine-grained information about where the model needs to be improved

Example: Testable Implications of DAGs



Example: Testable Implications of DAGs



This graph implies the following conditional independence relationships in the data:

$$M \perp\!\!\!\perp Z_2 | X, Z_1$$
$$X \perp\!\!\!\perp Z_3$$
$$Y \perp\!\!\!\perp Z_1 | M, X, Z_2, Z_3$$
$$Z_1 \perp\!\!\!\perp Z_2$$
$$Z_1 \perp\!\!\!\perp Z_3$$
$$Z_2 \perp\!\!\!\perp Z_3$$

Causal Discovery

- Causal discovery algorithms turn this idea on its head
  - Find all conditional independence relationships in the data
  - Construct the DAG that is compatible with these conditional independencies
- Remember the canonical d-separation relations

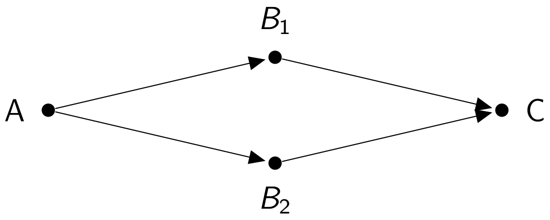
Chain:	$A \rightarrow B \rightarrow C$	$\Rightarrow$	$A \not\perp\!\!\!\perp C$ and $A \perp\!\!\!\perp C \mid B$
Fork:	$A \leftarrow B \rightarrow C$	$\Rightarrow$	$A \not\perp\!\!\!\perp C$ and $A \perp\!\!\!\perp C \mid B$
Collider:	$A \rightarrow B \leftarrow C$	$\Rightarrow$	$A \perp\!\!\!\perp C$ and $A \not\perp\!\!\!\perp C \mid B$

- We can empirically distinguish colliders from chain and forks and vice versa
- But we cannot distinguish chain and forks
- This means that we will often only be able to learn the underlying DAG up to a certain *equivalence class*

Causal Discovery – Assumptions

- Acyclicity
  - The underlying model (or data generating process) is acyclic
- Causal sufficiency
  - There are no hidden (or latent) variables
- Causal faithfulness
  - D-separation implies certain conditional independence relationships, but the other way round is not necessarily true
  - Causal faithfulness assumes that the reverse is indeed true (Heinze-Deml et al., 2018)
- Linearity and Gaussian errors
  - This assumption is relevant for conditional independence testing
  - Can be relaxed later on

Causal Faithfulness



- According to the d-separation criterion we would expect $A \not\perp\!\!\!\perp C$ and $A \perp\!\!\!\perp C | B_1, B_2$
- But $A \not\perp\!\!\!\perp C$ does not necessarily have to be the case if the B_1 path and the B_2 path cancel each other out perfectly
  - In this case, we would find $A \perp\!\!\!\perp C$ and $A \perp\!\!\!\perp C | B_1, B_2$, and could thus not infer the correct graph anymore
- Causal faithfulness rules out these pathological cases

The PC algorithm

- Given acyclicity, causal faithfulness and causal sufficiency we can apply the PC algorithm to infer the causal structure compatible with the data
    - Named after its inventors Peter Spirtes and Clark Glymour (Spirtes et al., 2000)
- The PC algorithm proceeds in three steps
    1. Determine the skeleton of the graph
    2. Determine the v-structures
    3. Determine further edge orientations

Step 1: Determine the Skeleton

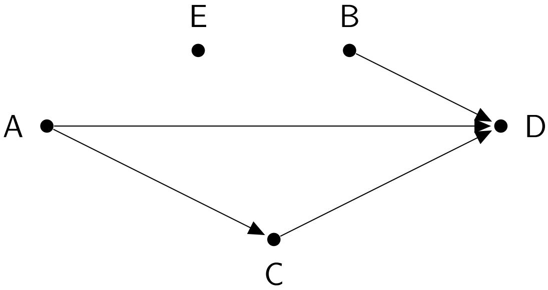
- The skeleton is the undirected graph that is obtained by replacing all directed edges with undirected edges
- Start with a complete undirected graph in which all nodes are connected
- For $k = 0, 1, 2, \ldots$ and two adjacent nodes i and j, test conditional independence of X_i and X_j given X_S for all $S \subseteq \text{adj}(i) \backslash \{j\}$ with $|S| = k$ and for all $S \subseteq \text{adj}(j) \backslash \{i\}$ with $|S| = k$
- Remove an edge if a conditional independence is found at the pre-specified significance level α
- Store the separating set S

Step 2: Determine the v-Structures

- Replace all edges by $\circ\!\!-\!\!\circ$
- Consider all unshielded triplets, i.e., $i \circ\!\!-\!\!\circ j \circ\!\!-\!\!\circ k$ where i and k are not adjacent
- Determine whether the triplet should be oriented as a v-structure (i.e., is a collider) based on the d-separation criterion

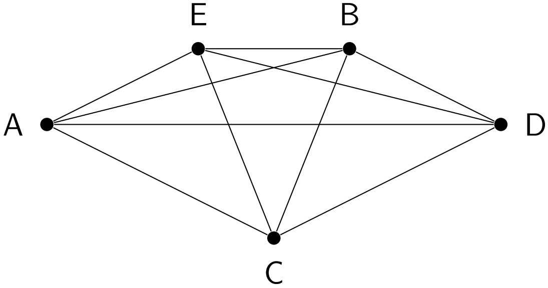
- Step 3: check for consistency across v-structures

Example: PC algorithm



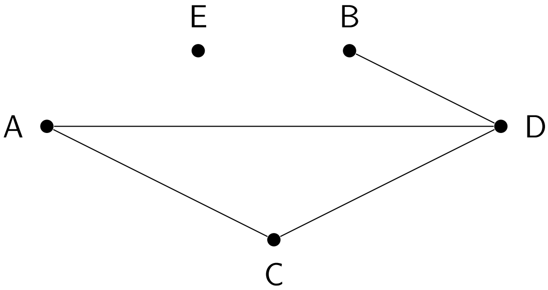
- Let us simulate data from the following DAG with linear causal relationships and Gaussian errors
- We can then easily run conditional independence tests via partial correlations

Example: PC algorithm (II)



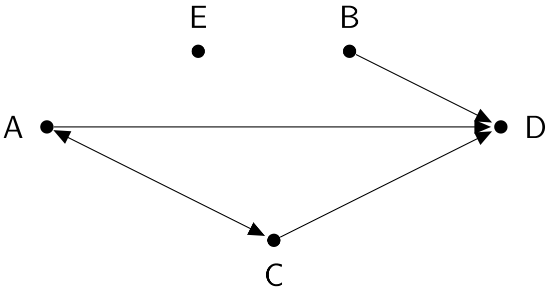
- To find the skeleton, we start with a complete undirected graph
- We find $A \perp\!\!\!\perp B$ and can therefore remove A—B
- We also find $A \perp\!\!\!\perp E$ and can therefore remove A—E
- And so forth...

Example: PC algorithm (III)



- Repeating this step for all possible variable combinations leads to the skeleton above
- Now we look at, e.g., $A \circ\!\!-\!\!\circ D \circ\!\!-\!\!\circ B$ and find that $A \not\perp\!\!\!\perp B$, but $A \perp\!\!\!\perp B | D$
- We can thus infer that D is a collider, such that $A \to D \leftarrow B$
- And so forth...

Example: PC algorithm (IV)



- Repeating this process for all unshielded triplets leads to the following *partially directed acyclic graph* (PDAG; Heinze-Deml et al., 2018)
- Note that we were not able to direct the edge between A and C. This means that both $A \rightarrow C$ and $A \leftarrow C$ are compatible with the data and in practice we could not distinguish between the two because both graphs belong to the same *equivalence class*

Practical Considerations

- How informative is the estimated equivalence class?
- The computational burden of the algorithm increases exponentially in the number of nodes (Le et al., 2015)
- Conditional independence tests
  - Edges do not have to be linear and errors not Gaussian
  - Non-parametric kernel-based conditional independence tests for continuous variables (Zhang et al., 2012)
  - G^2 test for discrete variables
    - But curse of dimensionality poses a problem
  - Mixed data types add an additional layer of complexity (Tsagiris et al., 2018)

Practical Considerations & Further Readings

- Hidden variables
  - The PC algorithm assumes that we observe all variables in the true graph
  - The FCI algorithm – a variant of the PC algorithm – takes arbitrarily many hidden variables into account but is also less informative

- Further readings
  - Heinze-Deml et al. (2018)
  - Peters et al. (2017)
    - Free ebook available at: `https://mitpress.mit.edu/books/elements-causal-inference`
  - Spirtes et al. (2000)

References I

Heinze-Deml, C., Maathuis, M. H., , and Meinshausen, N. (2018). Causal structure learning. *Annual Review of Statistics and Its Application*, 5:371–391.

Le, T., Hoang, T., Li, J., Liu, L., Liu, H., and Hu, S. (2015). A fast pc algorithm for high dimensional causal discovery with multi-core pcs. In *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

Pearl, J., Glymour, M., and Jewell, N. P. (2016). *Causal Inference in Statistics: A Primer*. John Wiley & Sons Ltd, West Sussex, United Kingdom.

Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of Causal Inference*. The MIT Press.

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search*. The MIT Press, Cambride, MA, 2nd edition.

References II

Tsagiris, M., Borboudakis, G., Lagani, V., and Tsamardinos, I. (2018). Constraint-based causal discovery with mixed data. *International Journal of Data Science and Analytics*, 6:19–30.

Zhang, K., Peters, J., Janzing, D., and Schoelkopf, B. (2012). Kernel-based conditional independence test and application in causal discovery.