CH 4: GRAPHICAL MODELS

Prashant Shekhar, PhD
Assistant Professor of Data Science
Department of Mathematics
Embry-Riddle Aeronautical University, FL, USA
Email: shekharp@erau.edu

Outline

1. Graph refresher
2. Graph representation
3. Graphical Causal Models
4. Causality beyond DAGs

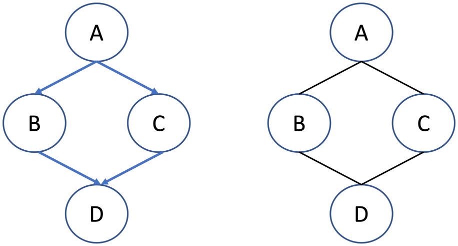
A REFRESHER ON GRAPHS

Graphs: Introduction

- You can think of them as discrete mathematical structures, abstract representations of real-world entities and relations between them, or computational data structures.
- What all of these perspectives have in common are the basic building blocks of graphs: **nodes** (also called vertices) and **edges** (links) that connect the nodes.
- Categories of graph:
  - Undirected versus directed
  - Cyclic versus acyclic
  - Connected versus disconnected
  - Weighted versus unweighted

Undirected vs Directed

- **Directed graphs** are graphs with directed edges, while **undirected graphs** have undirected edges. Figure below presents an example of a directed and undirected graph.



Directed Undirected

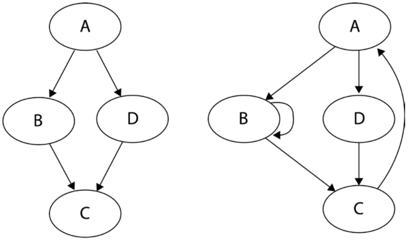
- As we saw in the previous lectures, we mark the edge direction with an arrow. Lines without arrows denote undirected edges. In the literature, we sometimes also see a line with two arrows (in both directions) to denote an undirected edge or to denote correlation rather than causation (for example, in structural equation models).

Undirected vs Directed

- When we know all the edges in the graph, but we are unsure about the direction of some of them, we can use **complete partially directed acyclic graphs (CPDAGs)** to represent such cases.
- CPDAGs are a special case of a broader class of **partially directed graphs**.

Cyclic vs Acyclic

- Cyclic graphs are graphs that allow for loops. In general, loops are paths that lead from a given node to itself. Loops can be direct (from a node to itself; so-called self-loops) or indirect (going through other nodes).



a b

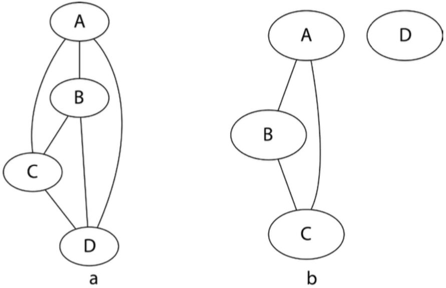
- Graph b here is a **cyclic graph** due to inherent loops. Graph a on the other hand is **acyclic**.

Cyclic vs Acyclic

- Most methods that we'll present here will assume that the underlying system can be accurately represented by an **acyclic graph**.
- However, there also exist causal methods that support cyclic relationships. We'll discuss an example of such a scenario in the last section of this lecture.

Connected versus disconnected

- In **connected graphs**, every node has an edge with at least one other node (for example, graph (a) below).



- A **fully-connected graph** contains edges between all possible pairs of variables.
- **Disconnected graphs** contain no edges.
- **Partially connected graph** has few connections with some variables unconnected (graph (b) above).

Weighted versus unweighted

- Weighted graphs contain additional information on the edges.
- Each edge is associated with a number (called a weight), which may represent the strength of connection between two nodes, distance, or any other metric that is useful in a particular case.
- In certain cases, weights might be restricted to be positive (for example, when modeling distance).
- Unweighted graphs have no weights on the edges; alternatively, we can see them as a special case of a weighted graph with all edge weights set to 1.
- In the context of causality, edge weights can encode the strength of the causal effect (note that it will only make sense in the linear case with all the structural equations being linear with no interactions).

Graph representation

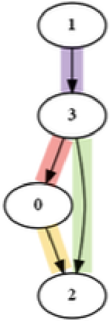
Graph representations

- We've seen enough graphs so far to understand how to present them visually.
- This representation is very intuitive and easy to work with for humans (at least for small graphs), but not very efficient for computers.
- To optimize certain types of computations, we can represent a graph as an **adjacency matrix**. This representation preserves the information about the graph structure and – possibly – the strength of connections between the nodes

Adjacency matrix examples

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$



Directed and acyclic

- Both adjacency matrices on the previous slide had $0's$ on diagonal.
- Both of these matrices represent valid DAGs, a type of graph with no cycles and with directed edges only.
- Any diagonal entry in a matrix will have an index in the (i, i) form, denoting an edge from node i to itself.

Graphical Causal Models (GCMs)

Graphical Causal Models (GCMs)

- We can define a graphical causal model as a set consisting of a graph and a set of functions that induce a joint distribution over the variables in the model.
- The basic building blocks of GCM graphs are the same as the basic elements of any directed graph: nodes and directed edges. In a GCM, each node is associated with a variable.
- Importantly,in GCMs, edges have a strictly causal interpretation, so that $A \mapsto B$ means that A causes B. Hence, the sheer graph structure is in some cases enough to decode information about statistical relationships between variables.
- In certain cases, we can infer the true causal graph structure by looking at the statistical relationships alone. This is possible when a special assumption called the **faithfulness assumption** is satisfied (will discuss later).

DAGs and causality

Pearl's definition

A causes B if B listens to A

In Pearlian terms, it means that if we change A, we also observe a change in B. The concept of a change in one variable leading to a change in another one is inherently related to the logic of interventions

- Visualize a simple DAG: $A \mapsto B \mapsto C$. Now imagine that we perform an intervention on node B and now the value of B entirely depends on our decision.
- In such a case, we expect that C will change its value accordingly, as C listens to B.
- On the other hand, under the intervention, A will no longer influence B, because our intervention fully controls the value of B. To reflect this in the graph, we remove the edge from A to B.
- The operation of removing the incoming edges from a node that we intervene upon is sometimes referred to as **graph mutilation**.

DAG (formally)

- DAG G consists of a set of vertices V and a set of directed edges E. Each node V_i is associated with a random variable X_i.
- Let's denote an edge from a vertex i to another vertex j as $i \mapsto j$. We call V_i a **parent** of V_j and we call V_j a **child** of V_i.
- For any directed path i $\mapsto \ldots \mapsto j$, with $k \in \mathbb{Z}_0^+$ (non-negative) vertices between i and j, we call V_i an **ancestor** of V_j and we call V_j a **descendant** of V_i.
- Note that because $k \geq 0$, parents are a special case of ancestors and children are a special case of descendants.
- By definition, in a DAG, there are no paths that start at vertex i that lead back to vertex i.

Problems with DAGs

An example

- When demand for product P grows, the producer might increase the supply in the hope of collecting potential profits from the market.
- Increased supply might cause a price drop, which in turn can increase demand further

Here increase in demand is informing further increase in demand, leading to an inherent cycle.

- Taking a more general perspective, causal DAGs in particular, and SCMs in general, have limitations.
- SCMs are an 'abstraction of underlying physical processes – abstraction whose domain of validity as causal models is limited'.
- That said, SCMs (but not DAGs) can be adapted to work with cycles