Advanced techniques in applied economics

Lecture 3: DAGs, interventions, and causal structure

Piotr Zwiernik

Universitat Pompeu Fabra Barcelona

Statistics, Probability and Machine Learning Research Group

BSE Barcelona School of Economics

Spring 2026

From undirected structure to causal structure

What changes today?

Today we move to **directed** graphs, where arrows encode asymmetric dependence and can be used to describe structural mechanisms.

Undirected graphs
Useful for summarizing conditional dependence networks.
They are symmetric and mainly descriptive.

Directed acyclic graphs
Useful for expressing directional or causal assumptions about how variables are generated.

Main question: Which variables matter for which others, and how would the system react to intervention?

Roadmap for today

1. DAGs and factorization
2. d-separation: chain, fork, collider
3. moralization as a way to test d-separation
4. Markov equivalence and CPDAGs
5. interventions and mechanism changes
6. back-door adjustment and causal identification

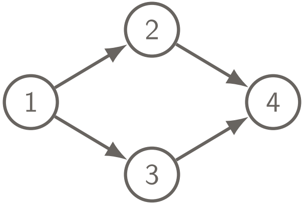
Memorable statement: a DAG is both a probabilistic model and a structural story.

Part 1: DAGs as structural models

What is a DAG?

A **directed acyclic graph** (DAG) consists of

- vertices $1, \ldots, m$ representing variables $X_1, \ldots, X_m$,
- directed edges $i \to j$,
- no directed cycles.



Interpretation

The arrow $i \to j$ says that X_i is a direct input into the mechanism generating X_j.

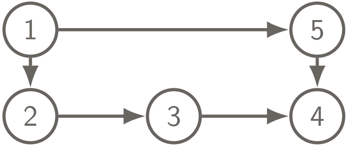
For standard definitions; see Chapter 3 in Højsgaard, Edwards, Lauritzen, Graphical models in R.

Parents, children, ancestors, descendants

Let D be a DAG.

For a node j:

- $\text{pa}(j)$ = parents of j,
- $\text{ch}(j)$ = children of j.
- $\text{an}(j)$ = ancestors of j,
- $\text{de}(j)$ = descendants of j.



For node 4:

$\text{pa}(4) = \{3, 5\}$, $\text{an}(4) = \{1, 2, 3, 5\}$.

For node 1:

$\text{ch}(1) = \{2, 5\}$, $\text{de}(1) = \{2, 3, 4, 5\}$.

Factorization over a DAG

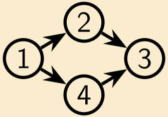
A distribution P is **Markov with respect to a DAG** D if its density factorizes as

$$p(x_1, \ldots, x_m) = \prod_{j=1}^{m} p\big(x_j \mid x_{\mathrm{pa}(j)}\big).$$

In a DAG, each factor involves one variable and its parents.

As in the undirected case, the graph has a dual role:

- it determines a factorization of the joint law,
- it implies conditional independence statements.



Then the joint law factorizes as:
$p(x_1, x_2, x_3, x_4) = p(x_1)\, p(x_2 \mid x_1)\, p(x_4 \mid x_1)\, p(x_3 \mid x_2, x_4)$.

For DAG factorization and the Markov property, see Lauritzen, *Graphical Models*, Chapter 3.

From DAGs to structural equations

A DAG can be viewed not only as a factorization of a joint law, but also as a system of structural assignments.

Structural equation view

For each node j, write

$$X_j = f_j\big(X_{\mathrm{pa}(j)}, \varepsilon_j\big), \qquad j = 1, \dots, m,$$

where the noise variables $\varepsilon_1, \dots, \varepsilon_m$ are jointly independent.

Why this matches the DAG

Each variable depends only on its parents and on its own noise term. So the arrows describe which variables enter directly into each mechanism.

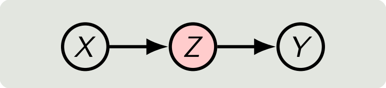
For the structural equation interpretation of DAGs, see Pearl, *Causality*, Chapter 1, or Peters, Janzing, and Schölkopf, *Elements of Causal Inference*, Chapter 5.

Part 2: d-separation

Three basic patterns

Everything starts with three local configurations.

Chain

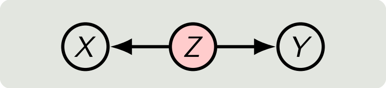


Always:
$p(x, y, z) = p(x)\, p(z \mid x)\, p(y \mid x, z)$

Here:
$p(x, y, z) = p(x)\, p(z \mid x)\, p(y \mid z)$

$X \perp\!\!\!\perp Y \mid Z$

Fork

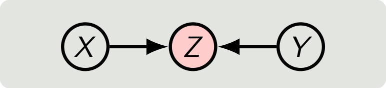


Always:
$p(x, y, z) = p(z)\, p(x \mid z)\, p(y \mid x, z)$

Here:
$p(x, y, z) = p(z)\, p(x \mid z)\, p(y \mid z)$

$X \perp\!\!\!\perp Y \mid Z$

Collider



Always:
$p(x, y, z) = p(x)\, p(y \mid x)\, p(z \mid x, y)$

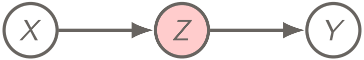
Here:
$p(x, y, z) = p(x)\, p(y)\, p(z \mid x, y)$

$X \perp\!\!\!\perp Y$

Two important notes

- Conditioning **blocks** a chain or fork, but **opens** a collider.
- Different DAGs may define the same model!

The chain pattern $X \perp\!\!\!\perp Y | Z$



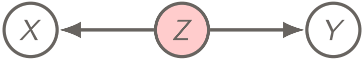
Key point

If we condition on Z, the path from X to Y is blocked, so $X \perp\!\!\!\perp Y \mid Z$.

Policy example
Policy X changes investment Z, which changes output Y. After fixing investment, policy adds no further predictive content for output.

Finance example
Monetary news X moves interest rates Z, which then affect stock returns Y. After fixing interest rates, monetary news adds no further predictive content for returns.

The fork pattern



Key point

If we condition on Z, the path from X to Y is blocked, so $X \perp\!\!\!\perp Y \mid Z$.

Demand-shock example

A common demand shock Z affects both price X and quantity Y. After controlling for the shock, the induced association disappears.

Ability example

Ability Z affects both education X and earnings Y. Conditioning on ability removes the spurious component of the association.

The collider pattern

Key point

If we do not condition on Z, the path from X to Y is blocked, so $X \perp\!\!\!\perp Y$.

Conditioning on Z removes that blockage.

Selection example
Admission Z to a program may depend on both test scores X and extracurricular achievements Y. Among admitted students, these may appear negatively related.

Labor-market example
Remaining in a demanding occupation Z may depend on both ability X and motivation Y. Conditioning on those who remain can induce dependence between them.

Conditional independence in a DAG is encoded by **d-separation**.

A path between nodes i and j is **blocked by** C if it contains

- a non-collider that belongs to C, or
- a collider that is neither in C nor has a descendant in C.

Definition: d-separation

Nodes i and j are **d-separated by** C if every path between i and j is blocked by C.

Proposition: d-separation gives conditional independence

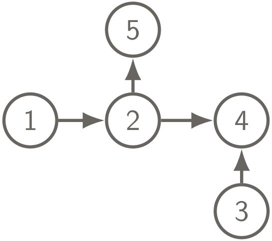
If a distribution is Markov with respect to the DAG and, for all $i \in A$ and $j \in B$, i and j are d-separated by C, then

$$X_A \perp\!\!\!\perp X_B \mid X_C.$$

For d-separation, see Pearl, *Causality*, Section 1.2.3.

A longer example

Consider the DAG



- 1 and 3 are d-separated by $\emptyset$ because the only path hits the collider 4.
- If we condition on 4, that path is no longer blocked, so 1 and 3 are not d-separated.
- 1 and 4 are d-separated by $\{2\}$ because the chain is blocked at 2.

Moralization: a practical way to test d-separation

A useful way to test d-separation is to convert the DAG into an undirected graph.

1. To test $X_A \perp\!\!\!\perp X_B | X_S$ keep only the ancestors of $A \cup B \cup S$.
2. **Moralize**: connect any two parents of a common child.
3. Drop all arrow directions.
4. Check whether S separates A and B in the resulting undirected graph.

Moralization criterion

A and B are d-separated by S in the DAG if and only if S separates A and B in the moralized ancestral graph.

For moralization and ancestral graphs in DAG separation, see Lauritzen, *Graphical Models*, Chapter 3.

Part 3: Markov equivalence and CPDAGs

Markov equivalence: why observational data do not identify every arrow

Different DAGs can encode the same set of conditional independence relations.

DAGs are **Markov equivalent** if they imply the same conditional independence statements.

Example

The following three DAGs are Markov equivalent. All of them imply $X_1 \perp\!\!\!\perp X_3 \mid X_2$.

The collider 

is **not** equivalent to the three above.

From observational data, we can typically recover only a Markov equivalence class.

Exact DAG identification may be possible under additional assumptions.

The Verma–Pearl characterization and CPDAGs

A **v-structure** is a triple of the form $X \to Z \leftarrow Y$ with no edge between X and Y.

Theorem (Verma–Pearl)

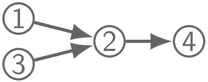
Two DAGs are Markov equivalent if and only if they have

- the same **skeleton**,
- the same **v-structures**.

A **CPDAG** represents an entire Markov equivalence class.

- **Directed edges:** directions shared by every DAG in the class.
- **Undirected edges:** directions that cannot be identified from observational data alone.

All v-structures are fixed across the equivalence class, but there may be more fixed arrows.



Above $1 \to 2 \leftarrow 3$ is a v-structure, but the arrow $2 \to 4$ is also fixed.

For the characterization of Markov equivalence, see Verma and Pearl (1990) or Chickering (1995).

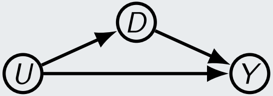
Part 4: Interventions and policy changes

Observing is not the same as intervening

Observational data tell us how variables move together. Policy analysis asks a different question:

What happens if we actively change one mechanism?

Simple setup



D = policy variable
Y = economic outcome
U = background conditions

Observational question: What is $\mathbb{P}(Y \mid D = d)$?

This is the outcome distribution among units with observed value $D = d$.

Interventional question: What is $\mathbb{P}(Y \mid \text{do}(D = d))$?

This is the outcome distribution if we actively set D to d.

In potential-outcomes language, $\mathbb{P}(Y \mid \text{do}(D = d))$ is the distribution of the counterfactual outcome $Y(d)$. If U affects both D and Y, then typically $\mathbb{P}(Y \mid D = d) \neq \mathbb{P}(Y \mid \text{do}(D = d))$.

For the do-operator and interventional distributions, see Pearl, *Causality*, Chapter 3.

Hard interventions: the *do* operator

A hard intervention replaces the mechanism generating one variable by an external assignment:

$$\mathrm{do}(D = d).$$

Graphical meaning

A hard intervention removes all arrows pointing into the intervened variable. The original mechanism for D is replaced by an external assignment.

The do-operator is the graphical version of a policy change: it tells us what the system would do if we externally set the treatment.

A tiny intervention example in R

Interventions in DAGs are not only a way to incorporate potential outcomes.
They can be used to identify the DAG.

Suppose the true system is: $D \to Y, \qquad Y = D + \varepsilon_Y.$

Observationally, association alone does not tell us whether D causes Y or vice versa. But under intervention on D, the response of Y becomes directly visible.

```
n <- 1000
# observational regime
d.obs <- rnorm(n)
y.obs <- d.obs + rnorm(n, sd = 0.5)
# intervention do(D = 2)
d.int <- rep(2, n)
```

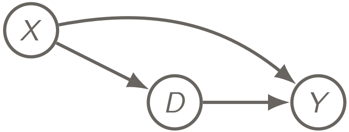
```
y.int <- d.int + rnorm(n, sd = 0.5)

mean(y.obs)
mean(y.int)
plot(d.obs, y.obs, pch = 16, cex = .4)
abline(v = 2, col = 2, lwd = 2)
```

Interpretation

Under do($D = 2$), the distribution of Y shifts. This is the interventional effect, not just an observational association.

Why observational comparisons can be misleading

Suppose we want to understand the effect of D on Y.



The variable X affects who receives treatment D, and it also affects the outcome Y.
So treated and untreated units may already differ before any effect of D is even considered.

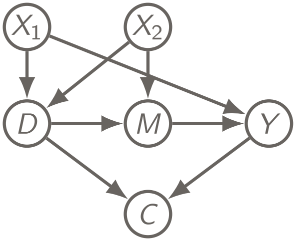
Main message

If we compare Y across treated and untreated units without taking X into account, the difference mixes: (i) the effect of D on Y, (ii) the pre-existing differences created by X.

In potential-outcomes language, $\mathbb{E}(Y \mid D = 1) - \mathbb{E}(Y \mid D = 0)$ need not equal $\mathbb{E}(Y(1) - Y(0))$.

Back-door paths and adjustment

A path from D to Y is called a **back-door path** if it starts with an arrow pointing into D.



Back-door criterion

A set of variables X is a valid adjustment set if

- no element of X is a descendant of D,
- X blocks every back-door path from D to Y.

Adjust for X_1, X_2

They are not descendants of D and they block the back-door paths

$D \leftarrow X_1 \rightarrow Y, D \leftarrow X_2 \rightarrow M \rightarrow Y$.

Do not adjust for M

M is a descendant of D. It is also a mediator on the directed path $D \rightarrow M \rightarrow Y$.

Do not adjust for C

C is a descendant of D. It is also a collider, so conditioning on it creates extra association between D and Y.

For the back-door criterion, see Pearl, *Causality*, Chapter 3.

What the back-door criterion gives us

Back-door criterion

If a set X blocks all back-door paths from D to Y, then the effect of intervening on D is identifiable from observational data by adjustment.

Connection to ignorability

In potential-outcomes language, this corresponds to **conditional ignorability**: $Y(d) \perp\!\!\!\perp D \mid X$. So the DAG gives a graphical route to deciding which covariates should be adjusted for.

Adjustment formula

$$\mathbb{P}(Y \mid \mathrm{do}(D = d)) = \sum_x \mathbb{P}(Y \mid D = d, X = x)\mathbb{P}(X = x).$$

Thus, the graph does not only describe dependence. It tells us when ordinary covariate adjustment is enough to recover the causal effect.

For the adjustment formula and the back-door criterion, see Pearl, *Causality*, Chapter 3.

A regression reading of adjustment

Suppose the causal effect of D on Y is modeled linearly as

$$Y = \alpha D + g(X) + \varepsilon, \qquad \mathbb{E}(\varepsilon \mid D, X) = 0.$$

If X is a valid back-door adjustment set, then conditioning on X removes the confounding part of the association between D and Y.

Econometric reading

In a linear model, this is exactly the situation where regressing Y on D and the right controls X can consistently estimate the causal coefficient α.

Take-away

- A DAG factorizes the joint law into local conditional mechanisms.
- d-separation tells us which conditional independences the DAG implies.
- Different DAGs may be Markov equivalent, so observational data do not identify every arrow.
- Interventions modify mechanisms, so $\mathbb{P}(Y \mid D = d) \neq \mathbb{P}(Y \mid \mathrm{do}(D = d))$.
- In linear SEMs, the graph appears directly in the coefficient matrix through the zero pattern of the structural parameters.

Main lesson of today: a DAG is both a statistical model for conditional independence and a structural model for interventions.

Looking ahead

Next lecture

We move from **representing** causal structure to **learning** it from data.

What comes next?

- constraint-based learning from conditional independences,
- score-based and structural approaches,
- when observational data identify only an equivalence class,
- linear structural equation models as a parametric framework,
- how extra assumptions such as non-Gaussianity can help orient edges.

Today the DAG was assumed known. Next time the question becomes: how much of it can we recover from data, and what extra structure helps us orient edges?