Contents

1 Do-Calculus: A Tutorial on Causal Reasoning with Graphs

This tutorial introduces **do-calculus**, Pearl's formal system for reasoning about causal effects using directed acyclic graphs (DAGs). Do-calculus provides a set of graph-based rules that let us transform interventional queries like $p(y \mid do(x))$ into expressions we can estimate from observational data.

1.1 What You'll Learn

- The difference between observational conditioning and causal intervention
- How to represent causal assumptions using DAGs
- The concept of d-separation and how it determines conditional independence
- The three rules of do-calculus
- How to derive the back-door and front-door adjustment formulas
- Common pitfalls like collider bias

1.2 Prerequisites

- Basic probability theory (conditional probability, Bayes' rule)
- Familiarity with the potential outcomes framework (helpful but not required)

1.3 Table of Contents

1.4 1. Basic Objects and Notation

1.4.1 Variables

In causal inference, we typically work with:

- X: **Treatment/intervention variable** (drug exposure, gene knockdown, CRISPR perturbation)
- Y: **Outcome** (phenotype, gene expression, survival)
- Z, W: **Covariates** (cell state, batch, donor, disease severity)
- U: **Unobserved confounder** (latent factors, unmeasured biology)

1.4.2 Two Kinds of Probability Statements

Observational (seeing):

$$p(y \mid x) \quad \text{or} \quad p(y \mid x, z)$$

This is standard conditional probability: "Among samples where $X = x$, what is the distribution of Y?"

Interventional (doing):

$$p(y \mid do(x))$$

This means: "What would Y look like if we *forced* X to take value x, regardless of what would naturally cause X to be that value?"

1.4.3 The Critical Distinction

In general:

$$p(y \mid do(x)) \neq p(y \mid x)$$

Why? Conditioning doesn't break confounding, but interventions do.

Example: Consider the relationship between carrying a lighter and lung cancer. Observationally, $p(\text{cancer} \mid \text{lighter})$ is elevated because smokers carry lighters. But $p(\text{cancer} \mid do(\text{lighter}))$—the effect of *forcing* someone to carry a lighter—is essentially zero. The intervention breaks the confounding path through smoking.

1.4.4 Graph Language: Directed Acyclic Graphs (DAGs)

We represent causal assumptions with a **directed acyclic graph (DAG)**:

- **Nodes** represent variables
- **Directed edges** ($\rightarrow$) represent direct causal influence
- **Bidirected edges** ($\leftrightarrow$) represent unobserved common causes (shorthand for a hidden U causing both endpoints)
- **Acyclic** means no variable can cause itself through any path

Example DAG:

```
  Z
 / \
v   v
X → Y
```

Here Z is a **confounder**: it causes both X and Y, creating a spurious association between them.

1.5 2. The Purpose of Do-Calculus

Do-calculus addresses two fundamental tasks:

1. **Identification:** Can we rewrite $p(y \mid do(x))$ using only observational quantities like $p(y \mid x, z)$, $p(z \mid x)$, etc.?

2. **Estimation:** Once identified, estimate the causal effect from data.

Do-calculus solves task (1) using graph-based rules. If you cannot rewrite the interventional query in observational terms, then your assumptions and measurements are insufficient to identify the causal effect from observational data alone—you need experiments.

1.6 3. Graph Surgery: The Do-Operator

The $do(x)$ operator has a precise graphical interpretation: **cut all arrows into X, then set $X = x$.**

1.6.1 Notation

- G: the original causal graph
- $G_{\overline{X}}$: the **mutilated graph** with all incoming edges to X removed ("bar X")
- $G_{\underline{X}}$: the graph with all outgoing edges from X removed ("underline X")

1.6.2 Intuition

When we intervene on X: * We sever X from its natural causes * X becomes like a randomized treatment—its value is set externally * All downstream effects of X remain intact

Example:

Original graph:

```
Z → X → Y
```

After $do(x)$ (graph $G_{\overline{X}}$):

```
Z   X → Y
```

The arrow $Z \to X$ is cut. Now X is independent of Z in the mutilated graph.

1.7 4. D-Separation: The Engine of Causal Reasoning

D-separation (directional separation) is the key concept that determines which conditional independences hold in a DAG. It's the foundation for all do-calculus rules.

1.7.1 Definition

A set of variables S **d-separates** A from B in a DAG G if S blocks every path between A and B.

1.7.2 The Three Path Types

To understand d-separation, we need to understand three types of path structures:

1.7.2.1 1. Chain (Mediation): $A \to M \to B$

```
A → M → B
```

- **Unconditionally:** A and B are associated (information flows through M)
- **Conditioning on M:** Blocks the path; $A \perp\!\!\!\perp B \mid M$

Example: Smoking → Tar in lungs → Cancer. Conditioning on tar blocks the association between smoking and cancer through this path.

1.7.2.2 2. Fork (Confounding): $A \leftarrow C \to B$

```
A ← C → B
```

- **Unconditionally:** A and B are associated (both caused by C)
- **Conditioning on C:** Blocks the path; $A \perp\!\!\!\perp B \mid C$

Example: Genetics → Smoking, Genetics → Cancer. Conditioning on genetics blocks the spurious association.

1.7.2.3 3. Collider (Inverted Fork): $A \to C \leftarrow B$

```
A → C ← B
```

- **Unconditionally:** Path is **blocked**; $A \perp\!\!\!\perp B$
- **Conditioning on C: Opens** the path; A and B become associated!

Example: Talent → Hollywood success ← Beauty. Among successful actors (conditioning on the collider), talent and beauty become negatively correlated—the "explain-away" effect.

1.7.3 D-Separation Algorithm

To check if S d-separates A from B:

1. List all paths between A and B
2. For each path, check if it's blocked by S:
   - A chain or fork is blocked if the middle node is in S
   - A collider is blocked if the collider (and all its descendants) is NOT in S
3. If ALL paths are blocked, A and B are d-separated given S

1.7.4 Worked D-Separation Examples

Example 1: Simple Confounding

```
   Z
  / \
 v   v
 X   Y
```

- Path $X \leftarrow Z \rightarrow Y$ is a fork at Z
- $X \perp\!\!\!\perp Y \mid Z$? **Yes** (conditioning on Z blocks the fork)
- $X \perp\!\!\!\perp Y$? **No** (path is open)

Example 2: Mediation

```
X → M → Y
```

- Path $X \rightarrow M \rightarrow Y$ is a chain through M
- $X \perp\!\!\!\perp Y \mid M$? **Yes** (conditioning blocks the chain)
- $X \perp\!\!\!\perp Y$? **No** (path is open)

Example 3: Collider

```
X → C ← Y
```

- Path $X \rightarrow C \leftarrow Y$ is a collider at C
- $X \perp\!\!\!\perp Y$? **Yes** (collider blocks the path)
- $X \perp\!\!\!\perp Y \mid C$? **No** (conditioning opens the collider!)

Example 4: Complex Graph

```
   U
  / \
 v   v
 X → M → Y
```

Paths from X to Y: 1. $X \rightarrow M \rightarrow Y$ (chain) 2. $X \leftarrow U \rightarrow Y$ (fork)

- $X \perp\!\!\!\perp Y$? **No** (both paths open)
- $X \perp\!\!\!\perp Y \mid U$? **No** (path 1 still open)
- $X \perp\!\!\!\perp Y \mid M$? **No** (path 2 still open)
- $X \perp\!\!\!\perp Y \mid M, U$? **Yes** (both paths blocked)

1.8 5. The Three Rules of Do-Calculus

Do-calculus provides three rules for manipulating interventional expressions. Each rule has a graphical condition based on d-separation in modified graphs.

1.8.1 Rule 1: Insertion/Deletion of Observations

$$p(y \mid do(x), z, w) = p(y \mid do(x), w)$$

Condition: $Y \perp\!\!\!\perp Z \mid X, W$ in $G_{\overline{X}}$ (the graph with incoming edges to X removed)

Intuition: After intervening on X, observing Z provides no additional information about Y (given W).

1.8.2 Rule 2: Action/Observation Exchange

$$p(y \mid do(x), do(z), w) = p(y \mid do(x), z, w)$$

Condition: $Y \perp\!\!\!\perp Z \mid X, W$ in $G_{\overline{X},\underline{Z}}$ (incoming edges to X removed, outgoing edges from Z removed)

Intuition: Under certain graphical conditions, "setting Z" is equivalent to "conditioning on Z" for predicting Y.

1.8.3 Rule 3: Insertion/Deletion of Actions

$$p(y \mid do(x), do(z), w) = p(y \mid do(x), w)$$

Condition: $Y \perp\!\!\!\perp Z \mid X, W$ in $G_{\overline{X},\overline{Z(W)}}$ where $Z(W)$ denotes Z-nodes that are not ancestors of any W-node in $G_{\overline{X}}$

Intuition: Intervening on Z doesn't affect Y once we've intervened on X and conditioned on W.

1.8.4 Using the Rules

These rules may seem abstract, but they become concrete in worked examples. The key insight: **d-separation in modified graphs determines when you can simplify interventional expressions**.

1.9 6. Worked Example: Back-Door Adjustment

1.9.1 The Graph

```
    Z
   / \
  v   v
  X → Y
```

Z is a confounder: it causes both X (treatment) and Y (outcome).

Goal: Identify $p(y \mid do(x))$.

1.9.2 Derivation

Step 1: Law of Total Probability

$$p(y \mid do(x)) = \sum_z p(y, z \mid do(x))$$

Step 2: Factor the Joint

$$p(y, z \mid do(x)) = p(y \mid z, do(x)) \cdot p(z \mid do(x))$$

So:

$$p(y \mid do(x)) = \sum_z p(y \mid z, do(x)) \cdot p(z \mid do(x))$$

Step 3: Simplify $p(z \mid do(x))$

In the mutilated graph $G_{\overline{X}}$, the edge $Z \to X$ is cut. Since Z has no path from X in $G_{\overline{X}}$:

$$p(z \mid do(x)) = p(z)$$

Step 4: Simplify $p(y \mid z, do(x))$

In $G_{\overline{X}}$, conditioning on Z blocks the back-door path. By Rule 2 (action/observation exchange):

$$p(y \mid z, do(x)) = p(y \mid x, z)$$

Step 5: Combine

$$\boxed{p(y \mid do(x)) = \sum_z p(y \mid x, z) \cdot p(z)}$$

This is the **back-door adjustment formula**. It expresses the causal effect entirely in terms of observational quantities.

1.9.3 Back-Door Criterion (General)

A set Z satisfies the **back-door criterion** relative to (X, Y) if:

1. No node in Z is a descendant of X
2. Z blocks every path between X and Y that contains an arrow into X

If Z satisfies the back-door criterion:

$$p(y \mid do(x)) = \sum_z p(y \mid x, z) \cdot p(z)$$

1.10 7. Worked Example: Front-Door Adjustment

The front-door adjustment is remarkable: it allows causal identification **even with unmeasured confounding** between treatment and outcome.

1.10.1 The Graph

```
    U (unobserved)
   / \
  ?   ?
  X → M → Y
```

Where: * U is an unobserved confounder between X and Y (shown as $X \leftrightarrow Y$) * M is a mediator: $X \to M \to Y$ * All causal effect of X on Y goes through M * No confounding between X and M

Goal: Identify $p(y \mid do(x))$ despite the unmeasured confounder.

1.10.2 Derivation

Step 1: Marginalize over the Mediator

$$p(y \mid do(x)) = \sum_m p(y \mid m, do(x)) \cdot p(m \mid do(x))$$

Step 2: Simplify $p(m \mid do(x))$

Since M is caused by X and there's no confounding between them:

$$p(m \mid do(x)) = p(m \mid x)$$

Step 3: Simplify $p(y \mid m, do(x))$

Once we condition on M, X has no direct effect on Y (all effect goes through M). Under the front-door conditions:

$$p(y \mid m, do(x)) = p(y \mid do(m))$$

So:

$$p(y \mid do(x)) = \sum_m p(y \mid do(m)) \cdot p(m \mid x)$$

Step 4: Identify $p(y \mid do(m))$ **via Back-Door**

For the effect of M on Y, X serves as a valid back-door adjustment set:

$$p(y \mid do(m)) = \sum_{x'} p(y \mid m, x') \cdot p(x')$$

Step 5: Combine

$$\boxed{p(y \mid do(x)) = \sum_m p(m \mid x) \sum_{x'} p(y \mid m, x') \cdot p(x')}$$

This is the **front-door adjustment formula**—one of the most elegant results in causal inference.

1.11 8. Worked Example: The Collider Trap

This example illustrates why "just control for more variables" is dangerous in causal inference.

1.11.1 The Graph

```
X → C ← U → Y
```

Where: * C is a **collider**: both X and U cause C * U is unobserved and causes Y * There is no direct effect of X on Y

1.11.2 Analysis

Without conditioning on C: * Path $X \to C \leftarrow U \to Y$ is blocked at collider C * $X \perp\!\!\!\perp Y$ (no causal effect, no spurious association) * $p(y \mid x) = p(y)$

With conditioning on C: * Conditioning on collider C **opens** the path * X and U become associated (given C) * This creates spurious association between X and Y * $p(y \mid x, c) \neq p(y)$

1.11.3 The Lesson

Conditioning on a collider (or its descendants) can **create** bias where none existed. This is called **collider bias** or **selection bias**.

Real-world example: Suppose C = "admitted to hospital", X = "has flu", U = "has heart disease", Y = "mortality". Among hospitalized patients (conditioning on C), flu and heart disease become negatively correlated (explaining away), which can create spurious associations with mortality.

1.12 9. Applications in Computational Biology

In computational biology, common causal structures include:

1.12.1 Confounders

- **Cell state** confounds perturbation effects (cycling cells transfect better AND express genes differently)
- **Batch/donor effects** confound treatment-outcome relationships
- **Disease severity** confounds treatment assignment and outcomes

1.12.2 Colliders (Selection Bias)

- **Cell survival** after perturbation: you only observe cells that survived, creating selection bias
- **Quality control filters**: cells passing QC may have correlated features

1.12.3 Mediators

- **Pathway activation** mediates perturbation effects on phenotype
- **Transcription factor activity** mediates genetic variant effects

1.12.4 When Do-Calculus Helps

Do-calculus provides the formal framework to determine:

1. **Which covariates are safe to adjust for** (avoid colliders and their descendants)
2. **When a mediator can rescue identification** (front-door criterion)
3. **When observational data cannot identify the effect** (need experiments or additional measurements)

1.13 10. Practical Checklist

When you encounter an interventional query $p(\cdot \mid do(\cdot))$:

1. **Draw the DAG** you believe represents the causal structure

2. **Check for back-door adjustment:**

    - Find a set Z that blocks all back-door paths from X to Y
    - Ensure Z contains no descendants of X

- If found: $p(y \mid do(x)) = \sum_z p(y \mid x, z) \cdot p(z)$

3. **If back-door fails, check for front-door:**
   - Find a mediator M such that:
     - $X \to M$ with no confounding
     - All effect of X on Y goes through M
     - X blocks confounding for $M \to Y$
4. **If both fail:**
   - The effect may be **non-identifiable** from observational data
   - Consider: experiments, instrumental variables, proxy variables, or stronger assumptions
5. **Beware of colliders:**
   - Never condition on a collider (or its descendants) unless you have a specific reason
   - Selection/survival variables are often colliders

1.14 Summary

Do-calculus provides a complete, algorithmic approach to causal identification:

Concept	Key Idea
$do(x)$ operator	Graph surgery: cut incoming edges to X
D-separation	Determines conditional independence from graph structure
Rule 1	Insert/delete observations based on d-separation in $G_{\overline{X}}$
Rule 2	Exchange actions and observations
Rule 3	Insert/delete actions
Back-door	Adjust for confounders that block back-door paths
Front-door	Use mediator to identify effect despite unmeasured confounding
Collider bias	Conditioning on colliders creates spurious associations

1.15 Further Reading

- Pearl, J. (2009). *Causality: Models, Reasoning, and Inference* (2nd ed.)
- Peters, J., Janzing, D., & Schölkopf, B. (2017). *Elements of Causal Inference*
- Hernán, M. A., & Robins, J. M. (2020). *Causal Inference: What If*

1.16 Next Steps

The natural continuation is to work through **identification by hand** for a realistic biology DAG—for example, with perturbation (X), cell cycle (Z), batch (B), pathway activation (M), and survival/selection (S)—showing exactly where each do-calculus rule applies and where identification fails.