Contents

1 Structural Causal Models: A Complete Tutorial

This tutorial introduces **Structural Causal Models (SCMs)**, Pearl's formal framework for causal reasoning. SCMs provide a unified approach to the three levels of causation: association, intervention, and counterfactuals.

1.1 What You'll Learn

- What structural causal models are and why they matter
- The three levels of the causal hierarchy
- How to implement SCMs in Python
- Interventions and the do-operator
- Counterfactual reasoning via abduction-action-prediction
- Applications to computational biology

1.2 Prerequisites

- Basic probability theory
- Familiarity with DAGs (see `do-calculus.md`)
- Understanding of potential outcomes (see `estimating-treatment-effects.md`)

1.3 Table of Contents

1.4 1. What are Structural Causal Models?

1.4.1 Definition

A **Structural Causal Model (SCM)** is a tuple $\mathcal{M} = \langle U, V, F \rangle$ where:

- U = **Exogenous variables** (unobserved noise, external factors)
- V = **Endogenous variables** (observed variables in the system)
- F = **Structural equations** (functions defining how V are generated from U and other V)

1.4.2 Structural Equations

Each endogenous variable V_i is determined by a structural equation:

$$V_i := f_i(\mathrm{PA}_i, U_i)$$

where: * PA_i are the parents of V_i (other endogenous variables) * U_i is the exogenous noise for V_i * f_i is a deterministic function

Key insight: The `:=` symbol means "is determined by" (not "equals"). This is an **assignment**, not an algebraic equation.

1.4.3 Example: Simple Linear SCM

Consider a simple causal relationship: smoking (X) causes lung cancer (Y).

Structural equations:

$$X := U_X$$

$$Y := 2X + U_Y$$

where: * $U_X \sim \mathcal{N}(0, 1)$ (individual propensity to smoke) * $U_Y \sim \mathcal{N}(0, 0.5)$ (other factors affecting cancer)

Interpretation: * Smoking is determined by individual propensity * Cancer risk is determined by smoking (coefficient 2) plus other factors

1.4.4 SCMs vs Statistical Models

Aspect	Statistical Model	Structural Causal Model
Focus	Associations, predictions	Causal mechanisms
Equations	$Y = 2X + \epsilon$	$Y := 2X + U_Y$
Interpretation	Correlation	Causation
Interventions	Not defined	Well-defined (do-operator)
Counterfactuals	Not computable	Computable

The key difference: SCMs model the **data-generating process**, not just the data distribution.

1.5 2. The Three Levels of Causation

Pearl's **Ladder of Causation** describes three increasingly powerful types of causal reasoning:

1.5.1 Level 1: Association (Seeing)

Question: What is?

Query: $P(Y \mid X)$

Example: "What is the cancer rate among smokers?"

Computation: Observational data is sufficient

$$P(Y \mid X) = \frac{P(X, Y)}{P(X)}$$

Limitation: Cannot distinguish causation from confounding

1.5.2 Level 2: Intervention (Doing)

Question: What if we do?

Query: $P(Y \mid do(X))$

Example: "What would the cancer rate be if we forced everyone to smoke?"

Computation: Requires causal assumptions (DAG + do-calculus)

$$P(Y \mid do(X)) \neq P(Y \mid X) \text{ (in general)}$$

Key insight: Interventions break incoming causal arrows

1.5.3 Level 3: Counterfactual (Imagining)

Question: What if we had done?

Query: $P(Y_x \mid X', Y')$

Example: "Would this patient have survived if they had not smoked, given that they smoked and died?"

Computation: Requires full SCM (structural equations)

Key insight: Counterfactuals are **individual-level** statements, not population-level

1.5.4 The Hierarchy

```
Level 3: Counterfactuals
         ↑
         | (requires SCM)
         |
Level 2: Interventions
         ↑
         | (requires DAG)
         |
Level 1: Associations
         ↑
         | (requires data)
```

Important: You cannot answer Level 3 questions with only Level 2 tools, and you cannot answer Level 2 questions with only Level 1 tools.

1.6 3. Implementing SCMs in Python

1.6.1 Basic SCM Class

```python
from causalbiolab.scm import StructuralCausalModel, SCMVariable
from scipy import stats

# Define variables
variables = {
    'X': SCMVariable(
        name='X',
        equation=lambda u_x: u_x,
        parents=[],
        noise_dist=stats.norm(0, 1)
    ),
    'Y': SCMVariable(
        name='Y',
        equation=lambda x, u_y: 2*x + u_y,
        parents=['X'],
        noise_dist=stats.norm(0, 0.5)
    )
}
```

```
# Create SCM
scm = StructuralCausalModel(variables)

# Sample observational data
data = scm.sample(n_samples=1000)
```

1.6.2 Example: Confounded SCM

```
# Z -> X, Z -> Y (Z is confounder)
variables = {
    'Z': SCMVariable(
        name='Z',
        equation=lambda u_z: u_z,
        parents=[],
        noise_dist=stats.norm(0, 1)
    ),
    'X': SCMVariable(
        name='X',
        equation=lambda z, u_x: z + u_x,
        parents=['Z'],
        noise_dist=stats.norm(0, 0.5)
    ),
    'Y': SCMVariable(
        name='Y',
        equation=lambda x, z, u_y: 2*x + z + u_y,
        parents=['X', 'Z'],
        noise_dist=stats.norm(0, 0.5)
    )
}

scm_confounded = StructuralCausalModel(variables)
```

1.6.3 Visualizing the DAG

Every SCM induces a DAG:

```
dag = scm_confounded.get_dag()
# {'Z': ['X', 'Y'], 'X': ['Y'], 'Y': []}
```

1.7 4. Interventions and the Do-Operator

1.7.1 Graph Surgery Interpretation

The do-operator $do(X = x)$ means:

1. **Cut** all incoming edges to X in the DAG
2. **Set** $X = x$ (constant value)

This creates a **mutilated graph** $G_{\overline{X}}$.

1.7.2 Implementing Interventions

```
# Intervene: set X = 1.5
scm_do_x = scm.intervene({'X': 1.5})
```

```
# Sample from intervened distribution
data_do_x = scm_do_x.sample(1000)

# Compare observational vs interventional
print(f"E[Y | X=1.5] = {data['Y'][data['X'] > 1.4].mean():.3f}")  # Observational
print(f"E[Y | do(X=1.5)] = {data_do_x['Y'].mean():.3f}")  # Interventional
```

1.7.3 Why Interventions Matter

Observational:

$$P(Y \mid X) = \sum_Z P(Y \mid X, Z)P(Z \mid X)$$

Includes confounding through Z.

Interventional:

$$P(Y \mid do(X)) = \sum_Z P(Y \mid X, Z)P(Z)$$

Removes confounding (no $P(Z \mid X)$).

1.7.4 Example: Confounding Bias

```
# Observational: biased by Z
data_obs = scm_confounded.sample(1000)
obs_effect = data_obs['Y'][data_obs['X'] > 0].mean() - data_obs['Y'][data_obs['X'] <= 0].mean()

# Interventional: unbiased
scm_do_x1 = scm_confounded.intervene({'X': 1.0})
scm_do_x0 = scm_confounded.intervene({'X': 0.0})
data_do_x1 = scm_do_x1.sample(1000)
data_do_x0 = scm_do_x0.sample(1000)
causal_effect = data_do_x1['Y'].mean() - data_do_x0['Y'].mean()

print(f"Observational effect: {obs_effect:.3f}")  # Biased
print(f"Causal effect: {causal_effect:.3f}")  # True effect   2.0
```

1.8 5. Counterfactual Reasoning

1.8.1 The Three-Step Process

Counterfactual computation follows Pearl's **abduction-action-prediction** framework:

1.8.1.1 Step 1: Abduction Infer the exogenous variables U from observed data.

Given observed $(X = x, Y = y)$, solve for U:

$$U_X = x$$

$$U_Y = y - 2x$$

1.8.1.2 Step 2: Action

Modify the SCM by applying intervention $do(X = x')$.

Create mutilated SCM with $X := x'$ (constant).

1.8.1.3 Step 3: Prediction

Compute the query variable using the inferred U and modified SCM.

$$Y_{x'} = 2x' + U_Y = 2x' + (y - 2x)$$

1.8.2 Example: Coffee Counterfactual

Scenario: Joe drank coffee ($T = 1$) and stayed awake ($Y = 1$). Would he have stayed awake if he hadn't drunk coffee?

SCM:

$$Y := T \cdot U + (T - 1)(U - 1)$$

Computation:

```python
from causalbiolab.scm.counterfactuals import compute_counterfactual

# Observed: T=1, Y=1
# Query: What if T=0?
y_cf = compute_counterfactual(
    scm,
    observed={'T': 1, 'Y': 1},
    intervention={'T': 0},
    query='Y'
)

print(f"Counterfactual outcome: {y_cf}")  # Y_0 = 0
```

Interpretation: Joe would not have stayed awake without coffee.

1.8.3 Linear SCMs: Efficient Counterfactuals

For linear SCMs, counterfactuals are particularly simple:

```python
from causalbiolab.scm.counterfactuals import LinearSCM

# Linear SCM: X -> Y with Y = 2X + U_Y
scm_linear = LinearSCM(
    coefficients={'Y': {'X': 2.0}},
    noise_distributions={'X': stats.norm(0, 1), 'Y': stats.norm(0, 0.5)}
)

# Counterfactual: observed X=1, Y=3; what if X=2?
y_cf = scm_linear.counterfactual(
    observed={'X': 1, 'Y': 3},
    intervention={'X': 2},
    query='Y'
)

print(f"Y_{{X=2}} = {y_cf:.3f}")  # Should be   5.0
```

1.8.4 Counterfactual vs Interventional Queries

Query Type	Question	Requires
Interventional	$E[Y \mid do(X = x)]$	Population-level, DAG sufficient
Counterfactual	Y_x for individual	Individual-level, full SCM needed

Key difference: Counterfactuals use **observed** values to infer individual-specific U, then predict under intervention.

1.9 6. Connection to Other Frameworks

1.9.1 SCMs and Potential Outcomes

Potential outcomes framework (Rubin): * $Y_i(1)$, $Y_i(0)$ are potential outcomes * ATE $= E[Y(1) - Y(0)]$

SCM perspective: * Potential outcomes are **counterfactuals** * $Y_i(1) = f_Y(\text{PA}_i, U_i)$ when $T_i := 1$ * $Y_i(0) = f_Y(\text{PA}_i, U_i)$ when $T_i := 0$

Connection:

$$\text{ATE} = E[Y \mid do(T = 1)] - E[Y \mid do(T = 0)]$$

SCMs provide the **mechanism** underlying potential outcomes.

1.9.2 SCMs and Do-Calculus

Do-calculus provides rules for identifying $P(Y \mid do(X))$ from observational data.

SCMs provide the **implementation**: * DAG structure comes from SCM * Do-operator is graph surgery on SCM * Identification formulas compute expectations in mutilated SCM

Example: Back-door adjustment

$$P(Y \mid do(X)) = \sum_Z P(Y \mid X, Z) P(Z)$$

In SCM terms: 1. Sample Z from marginal (unaffected by intervention) 2. Sample Y from conditional given X and Z

1.9.3 SCMs and Propensity Scores

Propensity score: $e(X) = P(T = 1 \mid X)$

In SCM: * $e(X)$ emerges from structural equation for T * IPW reweights to simulate $do(T = t)$ * SCM makes explicit what IPW assumes

---

1.10 7. Biological Applications

1.10.1 Gene Regulatory Networks

SCM:

$$\text{TF} := U_{\text{TF}}$$

$$\text{Gene} := \sigma(\text{TF}) + U_{\text{Gene}}$$

$$\text{Protein} := \text{Gene} \cdot \exp(U_{\text{Protein}})$$

where $\sigma(x) = 1/(1 + e^{-x})$ is sigmoid activation.

Questions: * **Intervention:** What if we knock out the TF? $do(\text{TF} = 0)$ * **Counterfactual:** Would this cell express the gene if TF was higher?

```
from causalbiolab.scm.examples import gene_regulation_scm

scm_gene = gene_regulation_scm()

# Intervention: knockout TF
scm_knockout = scm_gene.intervene({'TF': 0})
data_knockout = scm_knockout.sample(1000)

print(f"Gene expression under knockout: {data_knockout['Gene'].mean():.3f}")
```

1.10.2 Drug Response Prediction

SCM:

$$\text{Genotype} := U_G > 0$$

$$\text{DrugMetabolism} := 0.5 \cdot \text{Genotype} \cdot \text{Dose} + U_M$$

$$\text{Response} := 2 \cdot \text{Dose} - \text{DrugMetabolism} + U_R$$

Counterfactual question: “Would this patient respond better with a different genotype?”

```
from causalbiolab.scm.examples import drug_response_scm

scm_drug = drug_response_scm()

# Observed: poor metabolizer (Genotype=1), low response
observed = {'Genotype': 1, 'DrugDose': 1.0, 'Response': 1.5}

# Counterfactual: what if normal metabolizer?
response_cf = scm_drug.counterfactual(
    observed=observed,
    intervention={'Genotype': 0},
    query='Response'
)
```

```
print(f"Counterfactual response: {response_cf:.3f}")
```

1.10.3 Cell Cycle Confounding

SCM:

$$\text{CellCycle} := U_{CC}$$

$$\text{Transfection} := \sigma(\text{CellCycle}) + U_T$$

$$\text{GeneExpression} := 2 \cdot \text{Transfection} + 0.5 \cdot \text{CellCycle} + U_G$$

Intervention: What if we control for cell cycle?

```
from causalbiolab.scm.examples import cell_cycle_confounding_scm

scm_cc = cell_cycle_confounding_scm()

# Observational: confounded
data_obs = scm_cc.sample(1000)

# Interventional: fix cell cycle
scm_fixed_cc = scm_cc.intervene({'CellCycle': 0})
data_fixed = scm_fixed_cc.sample(1000)

# Compare transfection effect
print("Observational correlation:", np.corrcoef(data_obs['Transfection'], data_obs['GeneExpression'])[0
print("Causal effect (CC fixed):", np.corrcoef(data_fixed['Transfection'], data_fixed['GeneExpression']]
```

1.10.4 Perturbation Response Prediction

Use case: Predict phenotype after CRISPR knockout

SCM approach: 1. Learn SCM from observational single-cell data 2. Intervene on target gene: $do(\text{Gene} = 0)$ 3. Predict downstream effects

Advantage over black-box models: Mechanistic interpretation, compositionality for multi-gene perturbations

---

1.11 8. Advanced Topics

1.11.1 Identifiability of Counterfactuals

Question: Can we compute counterfactuals from data?

Answer: Depends on the SCM structure.

Identifiable cases: * Linear SCMs with Gaussian noise * Monotonic functions with specific noise distributions * Discrete variables with finite support

Non-identifiable cases: * Nonlinear SCMs with arbitrary noise * Hidden confounders between treatment and outcome

Practical implication: For biology, often need to make **parametric assumptions** about structural equations.

1.11.2 Mediation Analysis

Question: How much of the effect goes through mediator M?

Natural Direct Effect (NDE):

$$\text{NDE} = E[Y_{X=1,M=M_0} - Y_{X=0,M=M_0}]$$

Natural Indirect Effect (NIE):

$$\text{NIE} = E[Y_{X=1,M=M_1} - Y_{X=1,M=M_0}]$$

where M_t is the mediator value under $X = t$.

Requires: Counterfactual reasoning (Level 3)

1.11.3 Fairness and Discrimination

Counterfactual fairness: A decision is fair if:

$$P(\hat{Y}_A \mid X, A = a) = P(\hat{Y}_{A'} \mid X, A = a)$$

for all a, a' (protected attributes).

Interpretation: Outcome would be the same if individual had different protected attribute.

Application: Ensure drug recommendations don't discriminate based on race/gender.

1.11.4 Model Explanation

Counterfactual explanations: "Your loan was denied because if your income were $10K higher, it would have been approved."

SCM approach: 1. Learn SCM from data 2. Compute counterfactuals for feature changes 3. Find minimal changes that flip prediction

---

1.12 Summary

1.12.1 Key Takeaways

1. **SCMs formalize causation** through structural equations
2. **Three levels of causation** require increasingly strong assumptions
3. **Interventions** break causal arrows (do-operator)
4. **Counterfactuals** require abduction-action-prediction
5. **SCMs unify** potential outcomes, do-calculus, and graphical models

1.12.2 When to Use SCMs

Use SCMs when you need: * Individual-level predictions (counterfactuals) * Mechanistic understanding (not just associations) * Composition of interventions (multi-gene knockouts) * Explanation of model predictions

Don't use SCMs when: * Only population-level effects needed (use potential outcomes) * Only identification needed (use do-calculus) * Structural equations unknown (use nonparametric methods)

1.12.3 Further Reading

- Pearl, J. (2009). *Causality: Models, Reasoning, and Inference*
- Pearl, J., & Mackenzie, D. (2018). *The Book of Why*
- Peters, J., Janzing, D., & Schölkopf, B. (2017). *Elements of Causal Inference*

1.12.4 Next Steps

1. **Interactive notebook:** Work through examples hands-on
2. **Biological applications:** Apply to gene networks, drug response
3. **Integration:** Connect SCMs to existing causal inference tools

---

1.13 Appendix: Mathematical Details

1.13.1 Formal Definition of SCM

An SCM $\mathcal{M} = \langle U, V, F, P(U) \rangle$ consists of:

- $U = \{U_1, \ldots, U_m\}$: exogenous variables
- $V = \{V_1, \ldots, V_n\}$: endogenous variables
- $F = \{f_1, \ldots, f_n\}$: structural functions where $V_i = f_i(\mathrm{PA}_i, U_i)$
- $P(U)$: joint distribution over exogenous variables

1.13.2 Interventions (Formal)

The **mutilated model** $\mathcal{M}_{\overline{X}}$ under $do(X = x)$ is:

$$\mathcal{M}_{\overline{X}} = \langle U, V, F_{\overline{X}}, P(U) \rangle$$

where $F_{\overline{X}}$ replaces f_X with constant function $f_X(\cdot) = x$.

1.13.3 Counterfactuals (Formal)

The **counterfactual** $Y_x(u)$ is the value of Y in model $\mathcal{M}_{\overline{X}}$ with exogenous values $U = u$:

$$Y_x(u) = f_Y^{\mathcal{M}_{\overline{X}}}(\mathrm{PA}_Y, U_Y)$$

evaluated recursively in topological order.

1.13.4 Twin Network

Counterfactuals can be visualized as a **twin network**: * Factual world: actual observations * Counterfactual world: intervened model * Shared exogenous variables U link the two worlds

This explains why counterfactuals are individual-specific: they depend on the specific U for that individual.