Week 5: Causal Discovery from Observational Data

MATH-516 Applied Statistics

Linda Mhalla

2025-03-17

Section 1

Introduction

Causal inference: What is the effect of the cause?

Aim: focuses on determining the effect of one variable on another, based on observational or experimental/interventional data, and a given set of assumptions

- **Purpose**: To estimate the effect of interventions or treatments
- **Methods**: Randomized trials, matching, regression, instrumental variables
- **Challenge**: Controlling for confounding variables

References:

- Pearl (2000), second edition (2009)
- Hernàn and Robins (2020)

Causal discovery: What is the cause of the effect?

Aim: uncovers the underlying causal structure of a dataset without necessarily having prior knowledge of the relationships between variables

- **Purpose**: To identify potential causal relationships among multiple variables
- **Methods**: To follow
- **Challenge**: Determining the direction of causation from observational data alone

References: To follow

Introduction: What to do when concerned with causal relationships?

Controlled experiments can be expensive, time-consuming, unethical, impractical, or even infeasible

Intriguing alternative: causal discovery from purely observational data Spirtes et al., 2000 and Pearl (2000), second edition (2009)

Question: Do observational data from $p(X, Y)$ suffice to identify the direction of causality between X and Y?

Answer: No, generally...

Introduction

Proposition (Non-unique Graphs)

For any joint distribution $p(X, Y)$, there exists a Structural Equation Model (SEM):

$$Y = g_Y(X, \epsilon_Y), \quad X \text{ is independent of } \epsilon_Y,$$

where g_Y is a measurable function and ϵ_Y is a random variable (noise)

Proof: Consider $g_Y(x, \epsilon) = F^{-1}_{Y|X=x}(\epsilon)$, where $\epsilon \sim \mathcal{U}(0, 1)$ and ϵ is independent of X

$\Rightarrow$ Every joint distribution $p(X, Y)$ admits SEMs in both possible causal directions, *unless restrictions are imposed on* g_Y and/or on the noise ϵ_Y

Introduction

Learn causal structure directly from statistical properties (e.g., conditional independencies) of observational data

- **Methods**:
  - **Constraint-based methods**: Identify conditional independencies and use them to infer causal structure
    - PC Algorithm or Fast Causal Inference (Spirtes, Glymour and Scheines, 2000)
  - **Restricted models methods**: Impose additional assumptions on model class, noise distributions, causal function
    - RESIT (Peters et al., 2014) or LiNGAM (Shimizu et al., 2006)
  - **Score-based methods**: Evaluate how well a causal model fits the data using a scoring function (penalized likelihood or log-posterior)
    - Greedy Equivalence Search (GES) of Chickering, 2003
- **Challenges**:
  - Underlying assumptions (e.g., no hidden confounders, correct model specification)
  - Computational complexity and feasibility in high-dimensional spaces
  - Reliability of conclusions drawn from purely observational data

Section 2

Definitions

Concepts of graphs

Definition (Graph)

A graph $\mathcal{G}$ is a pair $(\mathbf{X}, E)$ where:

- $\mathbf{X} = \{X_1, \ldots, X_p\}$ is the set of vertices
- $E \subseteq \{(X_i, X_j) \in \mathbf{X}^2 | i \neq j\}$ is the set of edges

A **path** is a sequence of edges such as $< (X_1 X_2)(X_2 X_3) \cdots (X_{k-1} X_k) >$

Definition (Directed Graph)

A directed graph $\mathcal{G}$ is a pair $(\mathbf{X}, E)$ where:

- $\mathbf{X} = \{X_1, \ldots, X_p\}$ is the set of vertices
- $E \subseteq \{(X_i, X_j)_{\rightarrow} \in \mathbf{X}^2 | i \neq j\}$ is the set of arcs. An arc $(X_i X_j)_{\rightarrow}$ is a directed edge from X_i to X_j

A **cycle** is a directed path from and to the same vertex, e.g.,

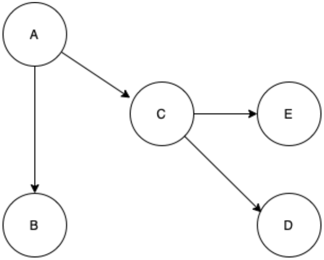
$$< (X_1 X_2)_{\rightarrow}, (X_2 X_3)_{\rightarrow}, \cdots, (X_k X_1)_{\rightarrow} >$$

Concepts of graphs

A directed acyclic graph (DAG) is a directed graph **without cycles**

The vertices of a DAG $\mathcal{G}$ represent random variables described by parent-child relationships:

- **Pa**$(i) = \{X_j : (X_jX_i)_{\rightarrow}$ is an edge of $\mathcal{G}\}$ (parent)
- **Ch**$(i) = \{X_j : (X_iX_j)_{\rightarrow}$ is an edge of $\mathcal{G}\}$ (child)
- **An**$(i) = \{X_j : X_j \rightarrow X_k \rightarrow \cdots X_l \rightarrow X_i$ is a directed path of $\mathcal{G}\}$ (ancestor)
- **De**$(i) = \{X_j : X_i \rightarrow X_k \rightarrow \cdots X_l \rightarrow X_j$ is a directed path of $\mathcal{G}\}$ (descendant)



Definition of d-separation

Given a path $X_1 X_2 \dots X_{m-1} X_m$, we say X_ℓ is a **collider** (relative to the path) if $X_{\ell-1} \to X_\ell \leftarrow X_{\ell+1}$.

The path is **blocked** by a set of nodes S with $X_1, X_m \notin S$ if for some $\ell \in \{2, \dots, m-1\}$ either:

- X_ℓ is not a collider and $X_\ell \in S$, or
- X_ℓ is a collider and neither X_ℓ nor any of its descendants are in S

Definition

Given disjoint subsets of nodes A, B and S, we say A and B are **d-separated** by S, and write:

$$A \perp\!\!\!\perp_{\mathcal{G}} B \mid S,$$

if every path between A and B is blocked by S

Structural Causal Models: The link between distributions and graphs!

A structural causal model (SCM) is a system of equations:

$$X_j := h_j(X_{P_j}, \varepsilon_j)$$

where:

- $\varepsilon_1, \dots, \varepsilon_p$ are independent noise variables
- $P_j \subseteq V \setminus \{j\}$ are such that the graph with $P_j = pa(j)$ is a DAG

Causal ordering: Given a DAG $\mathcal{G}$, a bijective mapping π

$$\pi : \{1, \dots, p\} \rightarrow \{1, \dots, p\}$$

is said to be a causal ordering of the variables if it satisfies

$$\pi(i) < \pi(j) \quad \text{if } j \in \mathbf{De}(i)$$

Definitions

- **Markov condition**: Conditional on the set of all its parent nodes, a node is independent of all other non-descendants nodes. Under the Markov condition, the joint probability satisfies

$$\Pr(X_1, X_2, \dots, X_p) = \prod_{i=1}^{p} \Pr(X_i \mid \mathbf{Pa}(i))$$

  $\Rightarrow$ connects a distribution and a causal DAG

- **Global Markov condition**: $A \perp\!\!\!\perp_{\mathcal{G}} B \mid S \implies X_A \perp\!\!\!\perp X_B \mid X_S$, for all disjoint sets of vertices A, B, and S

- **Markov equivalence class**: A class of DAGs that encode the same conditional independence relationships among a set of variables
  $\Rightarrow$ while graphs in this class may have different structures, they entail the same set of conditional independencies

Proposition (valid in a DAG that generates P)

If $X_j \not\!\perp\!\!\!\perp X_k \mid X_A$ for every $A \subseteq V \setminus \{j, k\}$, then either $X_j \to X_k$ or $X_j \leftarrow X_k$

Section 3

Constraint-based

The PC algorithm

The classic PC (Peter–Clark) algorithm was introduced by Spirtes, Glymour and Scheines (2000), Section 5.4.2. The algorithm assumes specific statistical assumptions

Assumptions: The key assumptions of the PC algorithm are:

- **Causal sufficiency**: There are no hidden confounders
- **Faithfulness**: Conditional independencies in the data imply conditional independencies in the DAG
- **Markov condition**: A variable is conditionally independent of its non-descendants given its parents in the DAG

The algorithm

- measures **conditional independencies** in the (observational) data
- searches for graph which satisfies the assumptions $\Rightarrow$ we only get identifiability up to the Markov equivalence class

The PC algorithm

Algorithm

1. Start with a fully connected **undirected** graph
2. Test for conditional independence among all pairs and subsets of variables, i.e., test $X \perp Y \mid Z$ for varying sets Z
3. Remove edges based on the outcomes of conditional independence tests
4. Continue until no more edges can be removed without violating the conditional independencies observed in the data
5. Apply orientation rules (v-structures, collider, etc.) to orient as many undirected edges as possible
6. Make final adjustments to prevent cycles and ensure the graph remains acyclic

The Rank PC algorithm, proposed by Harris and Drton (2013), is an extension of the PC algorithm and uses the rank-based Spearman correlation to perform the independence tests $\Rightarrow$ more robust to non-Gaussian data

The PC algorithm with Gaussian data

In the multivariate Gaussian setting, testing conditional independence is equivalent to testing for zero partial correlation:

- **Hypothesis testing**:
  - Null hypothesis $H_0 : \rho_{XY|S} = 0$
  - Alternative hypothesis $H_A : \rho_{XY|S} \neq 0$
- **Significance level** α:
  - Serves as a tuning parameter for the PC algorithm

Note I: performance of the algorithm depends on the validity of the assumptions but mostly on the correctness of the independence tests

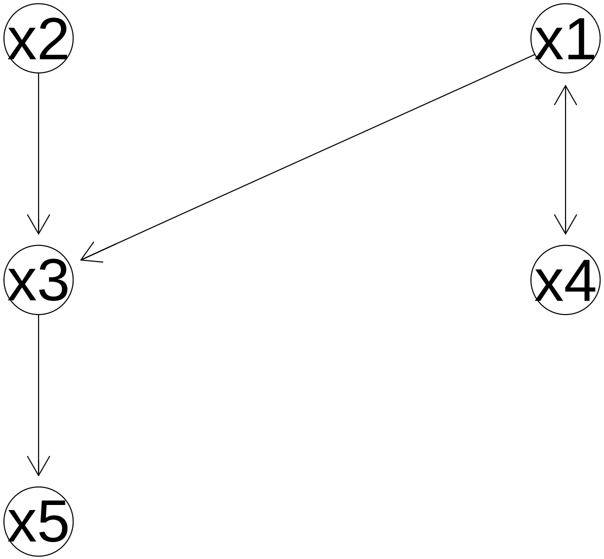
Consistency: Under the specified assumptions, if the conditional independence decisions are correct (e.g., use an oracle independence test), the PC algorithm is guaranteed to converge to the true Markov equivalence class

Note II: If Gaussianity assumed, conditional independencies $\iff$ non-zero coefs from a linear regression of X_i on X_{-i}. Generally, compute $\mathbb{E}\,\mathrm{Cov}(X, Y \mid Z)$ and use generalised covariance measure + double machine learning

The PC algorithm: Example

```
library(pcalg)
n <- 1000
set.seed(222)

eps1 <- rnorm(n,0,1)
eps2 <- rnorm(n,0,2)
eps3 <- rnorm(n,0,.7)
eps4 <- rnorm(n,0,1.5)
eps5 <- rnorm(n,0,1)

x1 <- eps1
x2 <- eps2
x3 <- x1 + x2 + eps3
x4 <- 2*x1 + eps4
x5 <- 3*x3 + eps5

data <- cbind(x1,x2,x3,x4,x5)
data <- data.frame(data)

pc.fit <- pc(suffStat = list(C = cor(data), n = n),
             indepTest = gaussCItest,
             labels = names(data),
             alpha = 0.01, verbose = FALSE)
plot(pc.fit@graph)
```

The PC algorithm: Example



Section 4

Restricted Models

Linear model with additive noise

Consider the linear structural equation model

$$\mathbf{X} = B\mathbf{X} + \varepsilon, \quad \text{with } B \in \mathbb{R}^{p \times p}, \mathbf{X} \in \mathbb{R}^p, \varepsilon \in \mathbb{R}^p$$

- Gaussian noise $\Rightarrow$ cannot distinguish between Markov equivalent graphs
  - different matrices B can generate the same distribution of X, which is Gaussian and completely determined by mean and covariance
- Non-Gaussian data
  - not fully determined by mean and covariance $\Rightarrow$ structure becomes identifiable

Linear model with additive Gaussian noise

Hypothesis: The cause X has a linear effect on Y. More precisely:

$$Y = \alpha X + \epsilon_Y, \quad \epsilon_Y \text{ is independent of } X,$$

where $\alpha \in \mathbb{R}$.

Theorem

Let X and Y be two continuous random variables whose joint distribution $p(X, Y)$ satisfies:

$$Y = \alpha X + \epsilon_Y, \quad \epsilon_Y \text{ is independent of } X$$

Then, there exists β in $\mathbb{R}$ and a continuous random variable ϵ_X such that:

$$X = \beta Y + \epsilon_X, \quad \epsilon_X \text{ is independent of } Y,$$

if and only if X and ϵ_Y are Gaussian

$\rightarrow$ It is sufficient that the noise or the effect is non-Gaussian to identify the causal direction

LiNGAM: Linear non-Gaussian acyclic models

LiNGAM assumes

- acyclicity $\Rightarrow$
  - a causal ordering exists (might not be unique)
  - the diagonal elements of B are zero (no self-loops)
  - using a causal ordering makes B strictly lower triangular
- ϵ is mean-zero non-Gaussian with positive variance
  - noise components are mutually independent, i.e., no hidden variables

Goal: Based on n i.i.d. observations of $\mathbf{X}$, estimate B or equivalently, learn the DAG (asnon-zeros in B are edges in the DAG)

LiNGAM: Linear non-Gaussian acyclic models

LiNGAM can be written as

$$\begin{aligned} \mathbf{X} &= B\mathbf{X} + \varepsilon \\ \mathbf{X} &= (I - B)^{-1}\varepsilon \end{aligned}$$

$\Rightarrow$ use Independent component analysis (ICA) to estimate/identify $(I - B)^{-1}$ and hence B

ICA model: $\mathbf{X} = A\mathbf{S}$

- $\mathbf{X} \in \mathbb{R}^p$: observed variables
- $\mathbf{S} \in \mathbb{R}^p$: independent and continuous non-Gaussian variables
- $A \in \mathbb{R}^{p \times p}$: unobserved full rank matrix

If $\mathbf{S}$ is non-Gaussian, then A is identifiable up to permutation, scaling and sign of the columns

$\Rightarrow$ Properties of B are exploited to find the “correct” permutation

ICA-LiNGAM algorithm

1. Given n i.i.d. observations of $\mathbf{X}$, use ICA to estimate $W = A^{-1} = (I - B)$ (up to permutation, scaling and sign of the columns of the "true A")
2. Find unique permutation of the rows of W that yields $\widetilde{W}$ without any zeros on the diagonal
3. Divide each row of $\widetilde{W}$ by its diagonal element to yield $\widetilde{W}'$ with only ones on the diagonal
4. Compute $\hat{B} = I - \widetilde{W}'$
5. Find causal order by making $\tilde{B} = \tilde{P}\hat{B}\tilde{P}^T$ as close as possible to strictly lower triangular (prune edge weights, e.g. using sparse regression or significance testin)

Note I: ICA-based algorithms may get stuck in local optima

Note II: Alternative method is DirectLiNGAM (guaranteed convergence as $n \to \infty$)

LiNGAM in R

```r
require(pcalg)
require(igraph)

### Linear SEM with uniform errors
n <- 1000

eps1 <- runif(n,-.5,.5)
eps2 <- runif(n,-.3,3)
eps3 <- runif(n,-.7,.7)
eps4 <- runif(n,-.5,.5)
eps5 <- runif(n,-1,1)

x1 <- eps1
x2 <- eps2
x3 <- x1 + x2 + eps3
x4 <- 3*x1 + eps4
x5 <- 2*x3 + eps5

data <- cbind(x1,x2,x3,x4,x5)
dataframe <- data.frame(data)

# apply LiNGAM
lingam.fit <- lingam(data)
round(lingam.fit$Bpruned,1)
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]  0.0    0  0.0    0    0
[2,]  0.0    0  0.0    0    0
[3,]  1.1    1  0.0    0    0
[4,]  3.0    0  0.0    0    0
[5,]  0.0    0  1.9    0    0
```

LiNGAM in R

```r
# Gaussian errors
n <- 1000

eps1 <- rnorm(n,0,1)
eps2 <- rnorm(n,0,2)
eps3 <- rnorm(n,0,.7)
eps4 <- rnorm(n,0,1.5)
eps5 <- rnorm(n,0,1)

x1 <- eps1
x2 <- eps2
x3 <- x1 + x2 + eps3
x4 <- 3*x1 + eps4
x5 <- 2*x3 + eps5

data <- cbind(x1,x2,x3,x4,x5)
dataframe <- data.frame(data)

# apply LiNGAM (although errors are Gaussian)
lingam.fit <- lingam(data)
round(lingam.fit$Bpruned,1)
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]  0.0    0  0.2    0    0
[2,] -0.9    0  0.9    0    0
[3,]  0.0    0  0.0    0    0
[4,]  3.0    0  0.0    0    0
[5,]  0.0    0  1.9    0    0
```

RESIT[1]: bivariate setting

RESIT stands for Regression with Subsequent Independence Test

Given variables X and Y, assume Y is caused by X and noise N through the additive noise model

$$Y = f(X) + N$$

The algorithm

1. fits a regression model (possibly non-linear) to estimate $\hat{f}(X)$ from X to Y
2. tests for independence between X and the residuals $\hat{N} = Y - \hat{f}(X)$ using an independence test like the Hilbert–Schmidt Independence Criterion (HSIC); see Appendix A.2 in Elements of Causal Inference
3. repeats the first two steps exchanging the roles of X and Y

If independence is not rejected for one direction and rejected for the other, infer the former as the causal direction

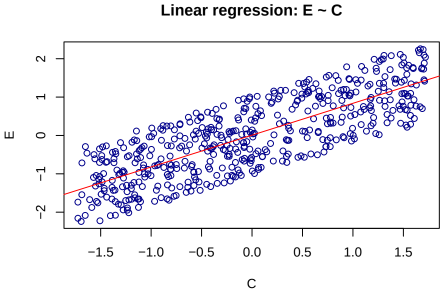
- Causality is assumed and algorithm finds the correct direction
- In practice, inferred causal direction is the one with the highest p-value

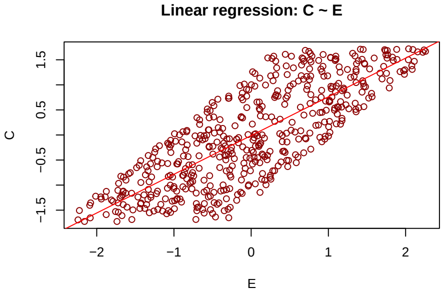
[1] see original paper

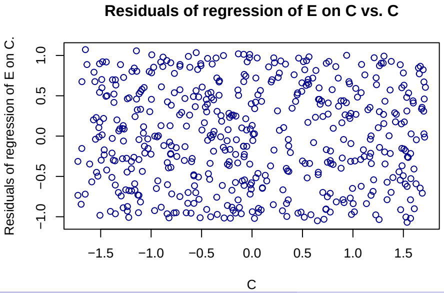
RESIT in R: Linear model with non-Gaussian noise

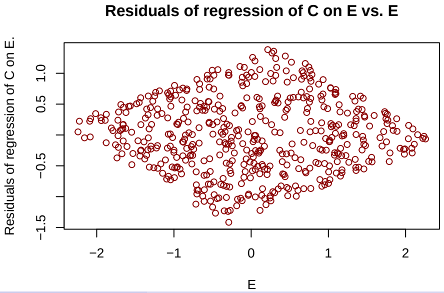
```r
library(dHSIC)
library(mgcv)
set.seed(1)

### RESIT: Regression with subsequent independence test
n <- 500

## Linear model with uniform noise
eps_cause  <- runif(n, -1.732051, 1.732051)
eps_effect <- runif(n, -1.03923, 1.03923)

C <- eps_cause
E <- 0.8*C + eps_effect

# fit model in forward direction
fit_causal <- lm(E~C)
residuals_causal <- fit_causal$residuals
# test whether residuals are independent of C
# dhsic.test(residuals_causal, C)$p.value # 0.9470529

# fit model in backward direction
fit_anticausal <- lm(C~E)
residuals_anticausal <- fit_anticausal$residuals
# test whether residuals are independent of E
# dhsic.test(residuals_anticausal, E)$p.value # 0.000999001
```

RESIT in R: Linear model with non-Gaussian noise





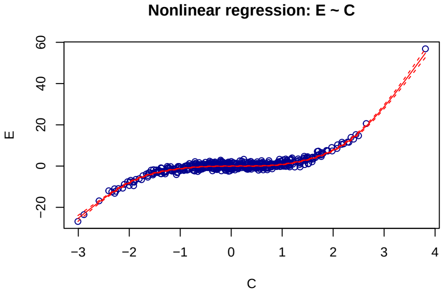


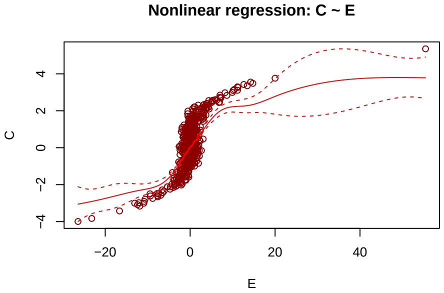


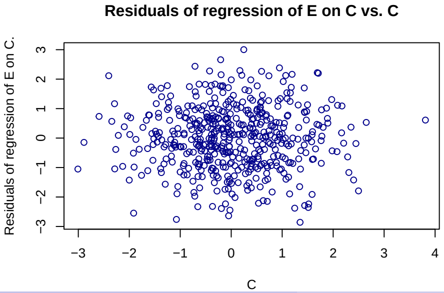
RESIT in R: Non-linear model with Gaussian noise

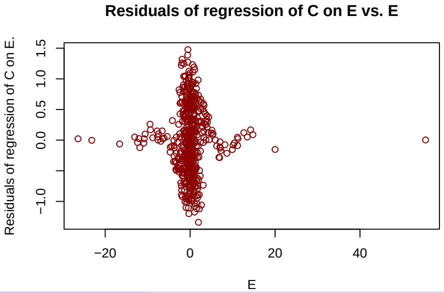
```r
## Nonlinear model with Gaussian noise
set.seed(1)
n <- 500

eps_cause <- rnorm(n)
eps_effect <- rnorm(n)

C <- eps_cause
E <- C^3 + eps_effect

# fit model in forward direction
fit_causal <- gam(E~s(C))
residuals_causal <- fit_causal$residuals
# test whether residuals are independent of C
# dhsic.test(residuals_causal, C)$p.value # 0.971029

# fit model in backward direction
fit_anticausal <- gam(C~s(E))
residuals_anticausal <- fit_anticausal$residuals
# test whether residuals are independent of E
# dhsic.test(residuals_anticausal, E)$p.value # 0.000999001
```

RESIT in R: Non-linear model with Gaussian noise





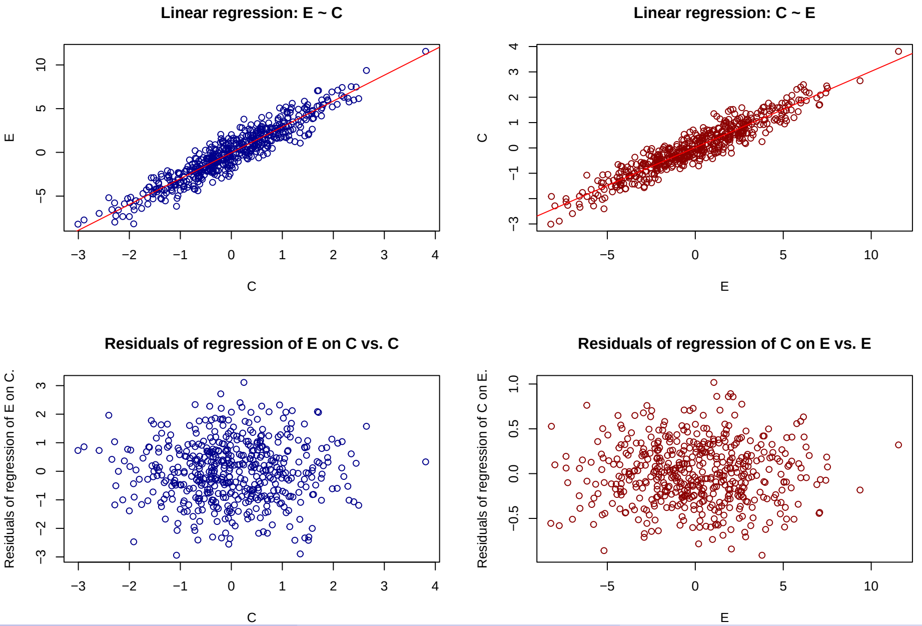




RESIT in R: Linear model with Gaussian noise

```r
## Nonlinear model with Gaussian noise
set.seed(1)
n <- 500

eps_cause <- rnorm(n)
eps_effect <- rnorm(n)

C <- eps_cause
E <- 3*C + eps_effect

# fit model in forward direction
fit_causal <- lm(E~C)
residuals_causal <- fit_causal$residuals
# test whether residuals are independent of C
# dhsic.test(residuals_causal, C)$p.value # 0.9330669

# fit model in backward direction
fit_anticausal <- lm(C~E)
residuals_anticausal <- fit_anticausal$residuals
# test whether residuals are independent of E
# dhsic.test(residuals_anticausal, E)$p.value # 0.8531469
```

RESIT in R: Linear model with Gaussian noise



RESIT: Extension to multivariate setting

Idea: Handling more than two variables by repeating bivariate tests

For each sink node X_i, it holds that $\epsilon_i \perp\!\!\!\perp \mathbf{X} \backslash \{X_i\}$

- Iterative approach for finding sink nodes (nodes with no children)

1. Regress each variable on all other remaining variables
2. Measure independence of residuals from the regression model with the regressors
3. The variable X_k with the least dependent residuals ϵ_k is deemed a sink node

- With the sink nodes isolated, a topological order and a connected DAG are obtained

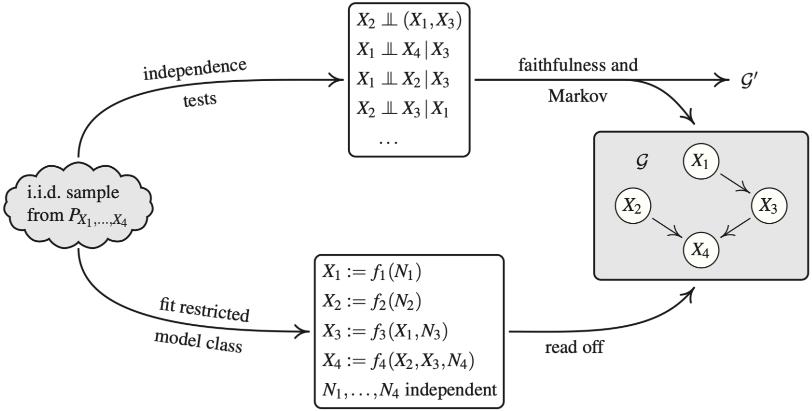
4. Remove superfluous edges based on the independence of the residuals
   - an edge from X_i to X_j is removed if the residuals $X_j - \hat{f}(\mathbf{X}_{Pa(j) \backslash i})$ are independent of $\mathbf{X}_{Pa(j) \backslash i}$

RESIT: Extension to multivariate setting

Note: Guarantees (consistency) for the performance of RESIT can be obtained under certain conditions (causal minimality)

Note: RESIT does not scale well to high dimensions as the step of edges' removal might depend on the order

Causal structure learning: summary



independence tests
$X_2 \perp\!\!\!\perp (X_1, X_3)$
$X_1 \perp\!\!\!\perp X_4 \mid X_3$
$X_1 \perp\!\!\!\perp X_2 \mid X_3$
$X_2 \perp\!\!\!\perp X_3 \mid X_1$
$\dots$
faithfulness and Markov
$\mathcal{G}'$
i.i.d. sample from $P_{X_1,\dots,X_4}$
$\mathcal{G}$
X_1
X_2
X_3
X_4
fit restricted model class
$X_1 := f_1(N_1)$
$X_2 := f_2(N_2)$
$X_3 := f_3(X_1, N_3)$
$X_4 := f_4(X_2, X_3, N_4)$
$N_1, \dots, N_4$ independent
read off

credit: Elements of Causal Inference

Section 5

References

Software packages & References

- Software packages/source code relevant to causal discovery and available online can be found here
- Review papers
  - Glymour, Zhang, and Spirtes (2019)
  - Mooij et al. (2016)
- Books
  - Peters, Janzing, and Schölkopf (2017)
  - Spirtes et al., 2000